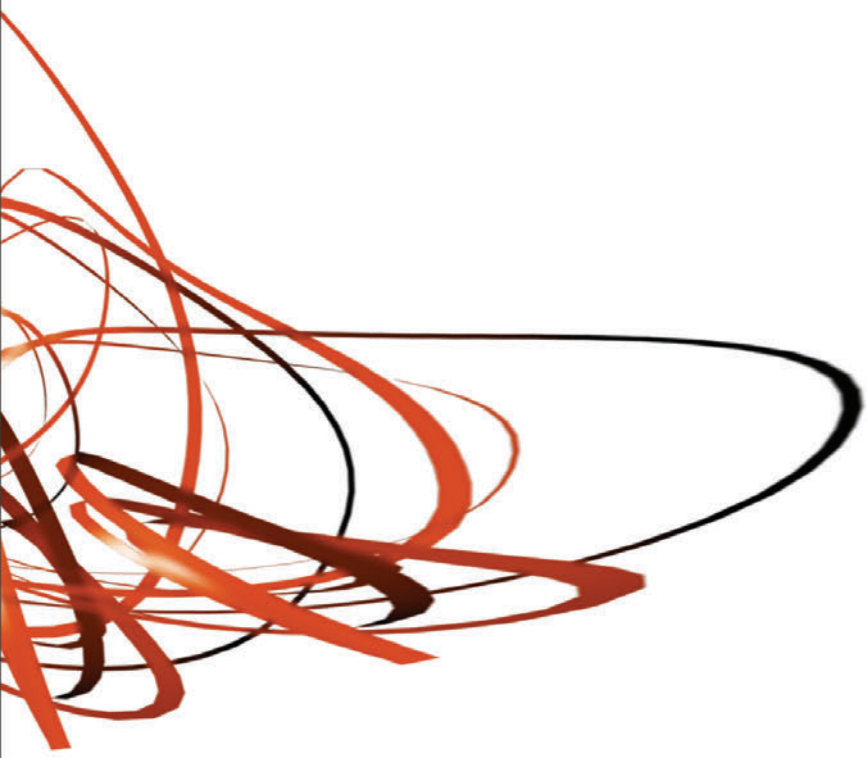
From Sequence to Structure And Back Again: An Alignment Tale

Simossis, V.A.

2005

FROM SEQUENCE TO STRUCTURE AND BACK AGAIN:
AN ALIGNMENT TALE

BY **VICTOR A SIMOSSIS**



FROM SEQUENCE TO STRUCTURE AND BACK AGAIN: AN ALIGNMENT TALE

Victor A. Simossis

VRIJE UNIVERSITEIT

FROM SEQUENCE TO STRUCTURE AND BACK AGAIN: AN ALIGNMENT TALE

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. T. Sminia,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op donderdag 7 juli 2005 om 10.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Victor-Athanasios Simossis

geboren te Athene

promotor: prof.dr. J. Heringa

IN MEMORY OF MRS. IWANNA PETA (1908-1998) AND CHRISTOS PASPALIDIS (1976-1998)

PhD committee:

- Dr. W.R. Taylor (London)
- Prof. Dr. P. Hogeweg (Utrecht)
- Prof. Dr. G. Vriend (Nijmegen)
- Prof. Dr. H.V. Westerhoff (VU)
- Dr. J. Kleinjung (VU)

The research described in this thesis was funded by the Medical Research Council (MRC) and the Faculty of Exact Sciences (FEW) of the Vrije Universiteit Amsterdam. The work was carried out at the MRC National Institute for Medical Research (NIMR), London and the Centre for Integrative Bioinformatics VU (IBIVU), Amsterdam, The Netherlands.

Published by:

Victor A. Simossis,
Bioinformatics Section,
Centre for Integrative Bioinformatics VU,
Vrije Universiteit Amsterdam (VU Amsterdam)

e-mail : vsimoss@cs.vu.nl
www : http://ibivu.cs.vu.nl, http://www.cs.vu.nl/~vsimoss

Book Cover Design by:

Anthony J.G. Kyriazis

Preface

"The most exciting phrase to hear in science,
the one that heralds new discoveries,
is not 'Eureka!' (I found it!), but 'That's funny....' "

- Isaac Asimov (1920 - 1992)

In early September of 1995 I left Greece to embark on a journey that was to me at the time, a purposeful adventure with a known destination: a BSc degree in Molecular Biology from the University of Edinburgh in Scotland. However, when I graduated, I realised that I had not reached my destination, but rather that I had simply stopped along the way to choose in which direction I would continue. My new destination became an MSc by Research degree in Cell Biology thanks to the generosity and motivation of Dr. Margarete M.S. Heck and soon enough progressed into a PhD in Bioinformatics.

Today, almost 10 years since my journey started, I realise that each step has been a journey in itself and not a destination *per se*. Each journey has been a unique learning process and not simply an academic achievement. The feeling of accomplishment, or in one word "Eureka!", is only overwhelming when there is an arduous journey that led up to it (Dr. M..M..S. Heck, Edinburgh, Scotland 1999). Indeed, I have found that all journeys worth taking in science begin with simple, everyday pauses where one says to oneself, "That's funny…".

This latest journey has been the longest and most demanding one for me yet. Over the last 4 years I am fortunate to have lived in two different countries; I have learned about their culture and their traditions; I have made many new friends. None of this would have been possible without the people that supported me in difficult times,

that gave me knowledge that cannot be found in any book and that have been a vital influence in my becoming the person I am today. I would like to acknowledge and thank them for their help and support in both completing this part of my journey and preparing me for the next step(s).

The PhD project described in this thesis was initiated in the Division of Mathematical Biology at the MRC National Institute for Medical Research (NIMR) in London and completed in the Centre for Integrative Bioinformatics VU (IBIVU) at the Vrije Universiteit Amsterdam. The work undertaken for this project was directly supervised by Professor Dr. Jaap Heringa in both locations.

First, I would like to thank my supervisor Professor Dr. Jaap Heringa for giving me the opportunity to carry out this research project in his group and Dr. Jens Kleinjung for his friendship and guidance throughout the last four years.

Jaap, you may not be the most organised of people, but which brilliant scientist is? Your enthusiasm and drive has been an inspiration to me and your humour, wisdom and council have lifted me in many difficult times. You, Helen and Alex have been a family away from home and I can only hope I have been worthy of your love. I came to you for a PhD in science and I have gained a mentor and a friend for life. What more could a student ask for?

Jens, we have shared an office for the last two years, we have talked about science and about life in general. You have been there for me in both scientific and personal difficulties. I want to thank you for your friendship and your guidance. I would also like to thank you and your father for your hospitality in Aachen. I hope you and Franca find what you truly wish for in your lives. “Ouuu ouuu!”.

Next, I would like to thank the people in the Mathematical Biology group in London and my friends at the NIMR, especially Ellie, Lucinda, Jenny, Costas, Babis, Despina, Sal, Grant and Letticia for making the two years we spent together unforgettable (very) (king), despite all the “pack of hydrophobic dog” events. Also, I would like to thank all the members of the IBIVU group and Maarten, Elena and Miriam for their friendship and scientific support on my many theories, especially the groundbreaking innovation in nutritional dynamics of the “going up stairs after lunch” theorem. A very special thanks is also due to Bart for his help in checking, binding and submitting the manuscript you will read later (or not!), while I was in Greece. Thank

you Bart and with my best wishes congratulations on your marriage this summer. Finally from the VU, I would like to thank Ruud for his kindness and incomparable zeal and the undergraduate and postgraduate students that tolerated my teaching.

From my Amsterdam friends, Wilco, Anne, Pierre, Frank, Raw-B, Jasper, Stavros, Bessy, Pepe, Floris, Joris, Bent, JJ, Mapi, Ilja, Zamani, Debbie and Robert you made Amsterdam a second home for me and thanks to you my time in Amsterdam has been one of the best periods of my life. Hopefully we will meet again at some point in the future to continue where we left off. “Figidy-fe-fellas, Haaaaidihooo!” A special thanks is also due to all the BC Schrobellaar club members for all our good times, genever projects, lengthy basketball tournaments and on-court adventures. Bravo! Bravo! I would love to thank you all one by one but I ask you: “Why? Why!? WHY!?”.

Saving the best for last, my undying gratitude and love goes to my family. My father Alvertos, my mother Antonia, my sister Vali, my aunt Stevi and my cousin Christos for their support, advice and love throughout my 28-year journey, so far. Without them at my side every step of the way I would have never come this far. Along side them, I want to thank my lifelong friends Manoli, George, Neek and Anthony for their continuous support and true friendship. Each one of you has moulded and affected me in your own special way and I only hope that this will continue in the many years to follow. Here is to our future adventures and journeys…

Finally, I want to thank Zoi for her love, her patience and unconditional support. Believing in someone is one of the greatest gifts a person can extend to another and you have done it without hesitation or demands. You have given me the best times of my life and I look forward to many more now that distance will soon no longer be part of our life. “Θα είναι καλή φάση… ξέρεις…πέτρα…”.

Amsterdam, January 2005

Victor A. Simossis

Contents

1. GENERAL INTRODUCTION 1

1.1 Research Objectives 7
1.2 Thesis Outline 8
1.3 Publications 10

2. AN OVERVIEW OF MULTIPLE SEQUENCE ALIGNMENT 13

2.1 Global and local alignment methods 16
2.2 Representing sequence and sequence block information 18
2.3 Performing multiple sequence alignment 23
2.4 Msa methodology 28
2.5 The Original PRALINE Method 34
2.6 Other State-of-the-art msa methods 36
2.7 Assessment of msa 40

3. SECONDARY STRUCTURE PREDICTION AND ITS CO-DEPENDENCE WITH MULTIPLE SEQUENCE ALIGNMENT 45

3.1 Secondary structure basics 46
3.2 Biochemical features of secondary structures used in prediction . 46
3.3 Secondary structure prediction: the beginning 48
3.4 From early to recent prediction: the key advances 49
3.5 Database searching and secondary structure prediction 51
3.6 State-of-the-art secondary structure prediction techniques 51
3.7 Evaluating secondary structure prediction methods 59
3.8 The interdependence of msa and secondary structure prediction ... 65

4. THE INFLUENCE OF GAPPED POSITIONS IN MULTIPLE SEQUENCE ALIGNMENTS ON SECONDARY STRUCTURE PREDICTION METHODS 69

4.1 Abstract 70
4.2 Introduction 70
4.3 Materials and Methods 73
4.4 Results 79
4.5 Discussion 92
4.6 Acknowledgements 94

5. A SIMPLE AND FAST SECONDARY STRUCTURE PREDICTION METHOD USING HIDDEN NEURAL NETWORKS 97

5.1 Abstract 98
5.2 Introduction 98

5.3 MATERIALS AND METHODS....100
5.4 RESULTS....104
5.5 DISCUSSION....109
5.6 ACKNOWLEDGEMENTS....111

6. HOMOLOGY-EXTENDED SEQUENCE ALIGNMENT....113
6.1 ABSTRACT....114
6.2 INTRODUCTION....114
6.3 MATERIALS AND METHODS....117
6.4 RESULTS....121
6.5 DISCUSSION....131
6.6 AVAILABILITY....133
6.7 ACKNOWLEDGEMENTS....133

7. IMPROVEMENT AND LIMITATIONS OF SECONDARY STRUCTURE-GUIDED MULTIPLE ALIGNMENT QUALITY....135
7.1 ABSTRACT....136
7.2 INTRODUCTION....136
7.3 MATERIALS AND METHODS....138
7.4 RESULTS....144
7.5 DISCUSSION....151
7.6 ACKNOWLEDGEMENTS....154

8. THE PRALINE SERVER....155
8.1 PROFILE PRE-PROCESSING AND ITERATION....156
8.2 HOMOLOGY-EXTENDED MULTIPLE ALIGNMENT....157
8.3 INTEGRATION OF SECONDARY STRUCTURE....157
8.4 THE PRALINE SERVER....158
8.5 THE OUTPUT PAGE....159
8.6 COLOUR SCHEMES....162
8.7 SUPPLEMENTARY MATERIAL....163
8.8 CAVEATS....165
8.9 CONCLUDING REMARKS....165
8.10 ACKNOWLEDGEMENTS....166

9. GENERAL DISCUSSION....167
9.1 REVIEWING THE KEY RESEARCH QUESTIONS....169
9.2 EPILOGUE....179

10. POSTFACE....181

11. BIBLIOGRAPHY....183

12. SAMENVATTING....193

13. SUMMARY....197

Chapter 1

General Introduction

"There is nothing permanent except change."

Heraklitus, 540-480 B.C.

"A wonderful harmony arises from joining together the seemingly unconnected."

Heraklitus, 540-480 B.C.

Regardless of the different levels of similarity between organisms in nature, we all originate from a very old group of common ancestors. It so happens that the evident variation between different species is the result of millions of years of gradual, but continual change (evolution). The fine-tuning mechanism that controls species evolution was studied, amongst others, by Charles Darwin (1809 –1882) who first formulated the theory of evolutionary selection. The theory states that variation occurs randomly and that the survival or extinction of each organism is determined by that organism's ability to adapt to its environment. In 1859 Darwin published his theories in one of the most revolutionising books to date, "The Origin of Species".

Today, almost 150 years later, the different visible characteristics of each organism, known as an organism's phenotype, are acknowledged to be the projection of distinct differences in the genes stored in every organism's genome (DNA). So ultimately, changes in the DNA of each organism, known as mutations, are the mechanism by which variation is achieved and the resulting ability of an organism to survive is the factor that determines which changes are first fixed and then preserved through evolution. As a result, the evolutionary pressure exerted on different regions of the genome depends on the importance of those regions for the survival of the organism. Mutations that have no effect on the fitness of an organism to survive will be preserved at a much higher rate than those that cause survival disabilities. Conversely, mutations that are fatal will never be preserved because they never enter the gene pool and therefore those genomic regions remain unaltered (conserved).

These principles are essential for the analysis of single genes or genomic regions. In particular, they form the basis of many computational methods that have been developed to analyse the huge amount of information that is being generated by the genome sequencing initiatives since before the turn of the millennium. More

importantly, since the first half of 2003, a complete first draft of the human genome has been made available to science and the use of computer-based sequence and structure analysis techniques have been instrumental in annotating the information. These techniques are part of a relatively recent marriage between computer science and biology called Bioinformatics, i.e. the use of biological rules and definitions to create computational methods for processing experimental data and providing guidelines for research carried out at the experimental level in research laboratories. Areas such as DNA sequence alignment, gene prediction, splice-site detection, single nucleotide polymorphism (SNP) detection and many more rely on the simple fact that functionally important regions of genes or gene clusters will be conserved enough through evolution, so that matching of genomic regions across species will help characterise them even with no prior knowledge about those regions. As a further step, the matching of unexplored genomic regions to related regions that are already annotated can help with the predictive inference of the structure or the function of these "unknown" regions.

If we now move one level up from DNA and look at the proteins that it encodes, we find that although DNA may change substantially, the amino acid composition of the proteins it encodes is relatively more conserved. The reason for this is that the nucleotide triplets that encode each amino acid are redundant, i.e. more than one triplet encodes the same amino acid. Since the ultimate fitness of an organism to survive depends on correct protein functionality, the evolutionary changes that occur in proteins due to DNA changes are a better source of information about which regions are functionally important. As a result, most of the research that involves the analysis of proteins and the detection of potential homologies (proteins that have the same common ancestor) is performed using protein information and not the DNA that encodes them. Such Bioinformatics research fields include sequence database searching (homology detection), protein sequence alignment, domain prediction (determining whether a protein is composed of a single or multiple domains), motif detection, i.e. the detection of (say) an active site pattern or a transcription factor binding motif, repeats detection and many more. From these fields, protein sequence alignment is currently perhaps the most commonly applied bioinformatics technique and has a history of over 30 years. It often leads to fundamental biological insight into sequence-structure-

function relationships of DNA or protein sequence families. In particular, multiple sequence alignment (MSA) is the central technique for inferring biological information from a set of more than two sequences. Much like the principles described above for DNA analysis, the information from homologous protein sequences allows the identification of conserved protein regions that may serve as key elements for predicting a protein's function by identifying functionally important residues (e.g. an active site or a ligand-binding motif). In addition, accurate comparison with homologues of known structure and/or function can also help in annotating and modelling un-characterised proteins.

In order to perform an alignment of two or more sequences (DNA or protein), an accurate representation of the evolutionary path that connects them through evolutionary time is needed. However, although we may have families of homologous proteins accurately aligned based on there three-dimensional structure, we still do not know when the actual changes occurred and therefore their order. This is important because although tracking a change from A to B may be adequately represented by taking the observed frequency of these events from the data we have available, there is no way to accurately take into account the order of the changes, i.e. A to B or B to A. As a result, all changes from A to B are inevitably considered equivalent to those from B to A. So, although extremely generalised, the frequency with which such evolutionary changes occur forms the basis for aligning two or more sequences together. Naturally, these few issues do not at all represent the whole story and considering the long and eventful history of the sequence alignment field an overview seems like an appropriate way to start this thesis and thus, one is provided in Chapter 2.

At this point, it is important to note that although the conversion of DNA into protein involves the generation of a string of amino acids, the final functional form of a protein requires the folding of random coil into a globular three-dimensional structure. In general, the folding process is thought to be encoded in the amino acid sequence (primary structure), although multiple pathways to the final folded form exist. For small proteins the folding process completes in one step, while in larger more complex proteins, first helices and strands are formed (secondary structure), then these elements fold into even more structured units (super-secondary structure) and finally all of it collapses into the protein's final functional form (tertiary structure). This latter form is

where the highest level of conservation exists and is the most reliable source of information for evolutionary and functional homology detection. At the experimental level, X-ray crystallography and Nuclear Magnetic Resonance (NMR) techniques are applied to solve protein three-dimensional structure, but they are much slower than the rate at which the information is produced by the various genome-sequencing initiatives. As a result, many Bioinformatics fields directly deal with solving the tertiary structure of proteins. To do this, the abundant primary structure information is used to predict the most probable secondary, super-secondary and tertiary structure conformations of a given protein. However, the success in these fields is inversely proportional to their complexity, i.e. secondary structure prediction is much more accurate than tertiary structure prediction (fold recognition). Consequently, apart from sequence information, most of the time secondary structure predictions are also used to guide the prediction of tertiary structures. So ultimately, the path from a protein's sequence to its folded structure involves the collection of homologous sequences for an accurate prediction of secondary structure that leads to the modelling of its possible folded states and then its probable function. On a genome-wide scale, a finely tuned and accurate version of this cascade would describe where each protein comes from, what it does and how it does it, thus providing key information for determining how an organism works.

The work described in this thesis involves two specific areas of Bioinformatics, namely protein sequence alignment and secondary structure prediction. Although important in their own respect, these two fields have also become increasingly interconnected in the last few years. Protein sequence alignment is now possibly one of the most important Bioinformatics fields and is the cornerstone for many evolving areas such as homology detection, secondary structure prediction, fold recognition and homology modelling. On the other hand, due to the large gap between protein sequence and structure availability, the use of secondary structure prediction methods has become an essential step in all of the latter areas, including protein sequence alignment. At present, these two areas are amongst the most essential applications of Bioinformatics in biological research and therefore advances in these fields have great implications in a number of related fields of biological research. Apart from the basic research of this thesis, which will be presented in the chapters to follow, the contributions that this project has offered are to a large extent the implementations of a

number of very useful tools that are beginning to be heavily used by the research community.

So why is the understanding of the evolutionary relationship between organisms and the mechanism of evolutionary change so important? The answer to this question is also the main reason for most biological research: the cure of disease. To do this, we need to study a disease and through our understanding of its biology try to find ways to prevent or cure it. This may involve the identification of a harmful agent such as the SARS virus, its mode of action and the cause of the pathogenic result. Alternatively, we may have to identify a mutation that causes a severe genetic disease such as cystic fibrosis (CF) or leads to aggressive forms of cancer. In all these cases, the most prominent targets for collecting information for the design of preventive measures and drug cures are the proteins that are affected. The identification of the functionally important regions of the proteins allows the design of repair, activation or de-activation strategies, depending on the situation. Therefore, accurate assessments of the regions that are highly conserved through evolution can give a good first indication of the function of these potentially unknown proteins and consequently focuses the cure strategies to specific target areas, such as the active site of a pathogenic enzyme. For these assessments to be accurate, the understanding of the evolutionary relationship between organisms and the mechanism of evolutionary change is essential.

Another important reason for determining how related organisms are, especially with respect to humans, is that research and testing of possible cures need to be applied to model systems in organisms that can be directly compared to humans. For example, the laboratory rat (*Ratus norvegicus*) has played a valuable role in efforts to understand human biology and to develop new and better drugs for nearly 200 years (Rat Genome Sequencing Project Consortium). Rat models have already helped to advance medical research in cardiovascular diseases (hypertension); psychiatric disorders (studies of behavioural intervention and addiction); neural regeneration; diabetes; surgery; transplantation; autoimmune disorders (rheumatoid arthritis); cancer; wound and bone healing; and space motion sickness.

Regardless of these few very specific examples, the collection of all the necessary information for studying a disease or the biology of a model organism involves a multi-disciplinary network of research fields cooperating at different levels.

In a world that is continually evolving, areas of science that were formerly considered seemingly unconnected have married and brought biological research into what is known as the post-genomic era.

1.1. RESEARCH OBJECTIVES

The main research objective of the work described in this thesis is the development of enhanced local weighting schemes for multiple sequence alignment in order to improve its ability to identify relationships between distantly related proteins. The challenge is that distant relationships are extremely hard to detect when only sequence information is taken into account and in the majority of cases the alignments are of very poor quality. This limits the ability of sequence alignment to provide accurate information for its various important applications, such as homology detection, homology modelling, sequence-function and sequence-structure determination.

The project has approached this problem by investigating the effects that homologous position-specific information collected from database searching and secondary structure information might have on alignment quality, when appropriately used. However, as mentioned in the section above, the available known protein structures are limited and therefore prediction methods need to be used to cover the gap when structures are not available. Accordingly, a number of secondary objectives arise and the questions they pose need to be addressed:

- Can we improve the use of alignment information for secondary structure prediction?
- Can prediction errors be limited by optimally combining the best predictions from a number of available state-of-the-art methods?
- Can the collection of homologous information from sequence databases improve alignment accuracy as has been shown for secondary structure prediction?
- Can the simultaneous use of extended homologous information and the resulting secondary structure predictions lead to an additive improvement?
- What are the types of errors in predicted secondary structure that limit alignment improvement capabilities?

- How do alignments affect secondary structure prediction accuracy and *visa versa*? What factors limit a smooth interdependence?
- Is the inter-dependence of multiple sequence alignment and secondary structure prediction a key aspect for designing a mutual optimisation scheme?

1.2. THESIS OUTLINE

The research described in this thesis is primarily an investigation of new ways in which multiple sequence alignment and secondary structure prediction can improve each other to open doors for further research into how sequence information can directly lead to the determination of a protein's function. The chapters of this thesis have been ordered so that they follow a logical path through the work, such that each chapter leads into the next research step as part of a chain of related events.

Since the start of this project, many advances have been made alongside our research. For the reader to have a more complete picture of the current status of the field and to place the research described in this thesis appropriately, Chapters 2 and 3 have been dedicated to the basic principles and an up-to-date history of the two fields.

In **Chapter 2**, an overview of multiple sequence alignment is presented, covering a history of nearly 30 years from the early pioneering methods to the current state-of-the-art techniques. Methodological and biological issues, end-user considerations as well as alignment evaluation issues are discussed. This chapter is based on (Simossis et al., 2003) and has been updated in appropriate places with facts that have come up since the original overview was published.

In **Chapter 3**, the background of secondary structure prediction and the key techniques that have been used in state-of-the-art methods are discussed. Its implications in sequence alignment, homology detection and the current evaluation and quality assessment standards are also discussed. The content of this chapter is mainly based on (Simossis and Heringa, 2004b; Simossis and Heringa, 2005b).

The research section of the thesis starts with **Chapter 4**, where we investigate the influence that gaps in multiple alignments have on the accuracy of state-of-the-art secondary structure prediction methods. We introduce a new dynamic programming method for optimising the segmentation of the predictions of a single prediction method, based on the input alignment and investigate which regions of the predicted

structures contain the most errors. This chapter is entirely based on the work published in (Simossis and Heringa, 2004b).

We continue in **Chapter 5** by describing YASPIN, a new secondary structure prediction method that was developed by combining an artificial neural network and a hidden Markov model, two of the currently most widely used machine learning techniques for secondary structure prediction. The results of the prediction error topology analysis in Chapter 4 formed the basis for the use of additional classifiers in the YASPIN method to separate the edges and cores of predicted secondary structure elements. We discuss YASPIN's advantages and compare it to other state-of-the-art methods where we find that YASPIN can predict strand elements much better than any other method. This chapter is entirely based on the work published in collaboration with Dr. K. Lin in the Mathematical Biology department of the MRC National Institute for Medical Research in London (Lin et al., 2005), to which I am joint first author.

In **Chapter 6** we move into the multiple sequence alignment section of the thesis and describe the PRALINE$_{PSI}$ extension to the existing PRALINE alignment tool, where multiple sequences are aligned by making use of the homologous information collected from independent sequence databases. The implications of this method as a possible standard addition to other alignment methods are discussed. The work described is entirely based on (Simossis et al., 2005).

As a concluding study to this research, in **Chapter 7** we integrate predicted secondary structure into multiple sequence alignment and investigate the effects of secondary structure prediction accuracy on the resulting alignments. In addition, we investigate the types of prediction errors that cause limitations in alignment improvement. The chapter is based on work currently submitted and under review (Simossis and Heringa, 2005a).

In **Chapter 8**, we give a comprehensive description of the online server we have created for the PRALINE and new PRALINE$_{PSI}$ alignment methods, how to use them and what they offer to the research community. This chapter is a merge between the original server article (Simossis and Heringa, 2003) and the new version, where the related work produced in this PhD is now also integrated (Simossis and Heringa, 2005c).

Finally, in **Chapter 9** we conclude the presentation of the research undertaken

in this PhD thesis by summarising the most important points and addressing the questions originally raised in this project. Finally, the implications we believe this research to have on the field and the possibilities for further work it has allowed are also discussed.

1.3. PUBLICATIONS

The published work that has resulted from the 4-year period from the 8th of January 2000, up to the 31st of March 2005 is listed below in ascending chronological order:

1. Simossis VA, Heringa J (2003) The PRALINE online server: optimising progressive multiple alignment on the web. Comput Biol Chem 27:511-519.

2. Simossis VA, Kleinjung J, Heringa J (2003) An overview of Multiple Sequence Alignment. In: Baxevanis AD (eds) Current Protocols in Bioinformatics. John Wiley, New York, 3.7.1-3.7.25.

3. Simossis VA, Heringa J (2004) Integrating protein secondary structure prediction and multiple sequence alignment. Curr Protein Pept Sci 5:249-266.

4. Simossis VA, Heringa J (2004) The influence of gapped positions in multiple sequence alignments on secondary structure prediction methods. Comput Biol Chem 28:351-366.

5. Lin K[§], Simossis VA[§], Taylor WR, Heringa J (2005) A simple and fast secondary structure prediction method using hidden neural networks. Bioinformatics 21:152-159.

6. Simossis VA, Kleinjung J, Heringa J (2005) Homology-extended sequence alignment. Nucleic Acids Res. 33(3): 816-824.

[§] Joint first authors

7. Simossis VA, Heringa J (2005) PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. Nucleic Acids Res., Vol. 33, Web Server issue, W1-W6.

8. Simossis VA, Heringa J (2005) Improvement and limitations of secondary structure-guided multiple alignment quality. Bioinformatics (submitted).

9. Simossis VA, Heringa J (2005) Local structure prediction of proteins. In: Xu Y, Xu D, and Liang J (eds). Computational methods for protein structure prediction and modeling. Springer Verlag, Chapter 8. (submitted)

Chapter 2

An Overview of Multiple Sequence Alignment

The majority of this chapter has been published in Simossis VA, Kleinjung J, Heringa J (2003) An overview of Multiple Sequence Alignment. In: Baxevanis AD (eds) Current Protocols in Bioinformatics. John Wiley, New York, 3.7.1-3.7.25.

A multiple sequence alignment (MSA) can be viewed as a two-dimensional table in which the sequences are the rows, and where the columns of equivalent amino acids have been arranged by placing gap characters in appropriate positions, such that the biological relationship of the sequences is represented best. It can provide a wealth of information about structure-function relationships within a set of protein sequences; such as the evolutionary conservation of functionally or structurally important amino acids at certain sequence positions or conserved hydrophobicity patterns in particular regions. It is also often useful as a starting point for site-directed mutagenesis experiments. In addition, as well as being an important means to glean the above biological clues by visual inspection, MSAs are an essential pre-requisite to many computational modes of analysis of protein families such as homology modelling, secondary structure prediction and phylogenetic reconstruction. They may further be used to derive profiles (Gribskov et al., 1987) or hidden Markov models (Haussler et al., 1993; Bucher et al., 1996) that can be used to scour databases for distantly related members of the family.

As a general rule, a MSA is an attempt to represent evolutionary related sequences in the most consistent way. Finding the maximal alignment score according to a given evolutionary model is equivalent to maximizing the probability that the sequences evolved as given by the MSA. Nonetheless, despite the considerable history of MSA (Needleman and Wunsch, 1970; Smith and Waterman, 1981; Hogeweg and Hesper, 1984; Barton and Sternberg, 1987; Corpet, 1988; Higgins and Sharp, 1988; Taylor, 1988; Lipman et al., 1989; Gotoh, 1993; Thompson et al., 1994; Gotoh, 1996; Stoye et al., 1997; Stoye, 1998; Heringa, 1999; Notredame et al., 2000; Heringa, 2002; Lee et al., 2002; Przybylski and Rost, 2002; Pei et al., 2003; Edgar, 2004; Do et al., 2005) the methodology is still under permanent development.

Many times a complicated relation exists between homologous sequences combined with a lack of information about their true evolutionary history and consequently, absolute certainty about the correctness of MSAs is often hard to achieve. Therefore, it is instructive to bear in mind that a computational biologist is dealing with 'truth' or 'correctness' at two levels: The first level is the biological reality (current and past), and the second level is the chosen model of this reality in terms of scoring schemes, graphs, or alignments. Thus, it is appropriate to quote: 'All models are

wrong but some are useful' (George E.P. Box, 1987). The fundamental model for all types of sequence comparisons is the (generalized) model of evolution. Such models have been derived from trusted sequence alignments by assuming a Markov model of evolution (Dayhoff et al., 1978; Muller and Vingron, 2000; Muller et al., 2002) or by empirical derivation (Gonnet et al., 1992; Henikoff and Henikoff, 1992; Jones et al., 1992; Benner et al., 1993). However, this generalized approach reflects a standardised evolutionary model and often introduce inconsistencies when applied to non-standard cases (Yu et al., 2003). A possible improvement has been recently suggested, where these generalized models are re-adjusted to better fit the evolutionary model of a specific organism or even the set of query sequences being aligned (Yu et al., 2003). In any case, the information about sequence evolution derived from any of these models is usually stored in amino acid substitution matrices. In the following sections we focus on higher-level aspects of modelling, concerning the complex relationships within sequence families, such as tree construction and progressive alignment strategies. These strategies are discussed along with practical considerations about the construction of meaningful MSAs.

With the completion of the first draft of the human genome (April 2003) and well over 300 genomes of other species, the accurate alignment of biological sequences has become more important than ever. This is due to the fact that the direct prediction of a protein's structure and function is still a major unsolved problem. To increase the knowledge of the function and interaction of protein sequences obtained by sequencing techniques, many initiatives are underway for large-scale proteomics and structure elucidation of novel genomic proteins. However, at present roughly half of the proteins in most sequenced species (60% in humans) do not have an assigned function, and consequently an important target of bioinformatics method development is aimed at gathering the function of an increased fraction of translated proteins by enhancing comparative sequence techniques and threading protocols. In the quest for knowledge about the role of a certain unknown protein in the cellular molecular network, comparing the query sequence with the many sequences in annotated protein sequence databases often leads to useful suggestions regarding the protein's 3-dimensional (3D) structure or molecular function. These suggestions are obtained by extrapolating the properties of sequences, residing in annotated public databases, which are identified as

"neighbours" of the query sequence by comparative sequence analysis techniques. During the last three decades, such homology search methods have arguably led to the putative characterization (annotation) of more sequences than any other single technology. Significant progress has been made in homology searching over the last few years by employing MSA techniques in iterative sequence database search strategies (Altschul and Koonin, 1998; Taylor, 1998; Taylor and Brown, 1999), the use of profiles to represent a query and the database information (Karplus et al., 1998; Jaroszewski et al., 2000; Yona and Levitt, 2002; Mittelman et al., 2003; Sadreyev et al., 2003; Capriotti et al., 2004; Edgar and Sjolander, 2004a; Soding, 2004; Tomii and Akiyama, 2004; Wang and Dunbrack, 2004) and as an increasingly popular approach, the integration of secondary structure information (Ginalski et al., 2003; Chung and Yona, 2004; Ginalski et al., 2004).

Since the advent of the genome sequencing projects and resulting rapid expansion of sequence databases, the method of indirect inference by comparative sequence techniques has only gained in significance. Many current research projects aim to improve the sensitivity of (multiple) sequence alignment techniques, which require high-performance computing given the current and rapidly growing database sizes. Given the plethora of sequence data, the alignment engines also have to be extremely fast and fully automatic to be included in genomic pipelines.

2.1. GLOBAL AND LOCAL ALIGNMENT METHODS

Many MSA techniques perform *global* alignment (Needleman and Wunsch, 1970) and match sequences over their full lengths. Problems with this approach can arise when sequences that are only homologous over local regions are compared. In such cases, global alignment techniques might fail to recognize highly similar internal regions because these may be overshadowed by dissimilar stretches and high gap penalties are normally required to achieve proper global matching. Moreover, many biological sequences are modular and show shuffled domains (Heringa and Taylor, 1997), which can render a global alignment of two complete sequences meaningless. The occurrence of varying numbers of internal sequence repeats (Heringa, 1998) can also severely limit the applicability of global methods. In general, when there is a large difference in the lengths of two sequences to be compared, it is advisable to include

local alignment techniques in the analysis. To address these problems, Smith and Waterman (Smith and Waterman, 1981) early on developed a so-called *local* alignment technique in which the most similar regions in two sequences are selected and aligned. The algorithm has been extended in various techniques to compute a list of top-scoring pair-wise local alignments (Waterman and Eggert, 1987; Huang et al., 1990). Alignments produced by the latter techniques are non-intersecting; i.e., they have no matched pair of amino acids in common. For multiple sequences, the main automatic methods include the Gibbs sampler (Lawrence et al., 1993), MEME (Bailey and Elkan, 1994) and Dialign2 (Morgenstern, 1999). These local MSA programs often perform well when there is a clear block of un-gapped alignment shared by all of the sequences but perform poorly, however, under moderate gap requirements and show inferior results over general sets of test cases when compared with global methods (Thompson et al., 1999b; Notredame et al., 2000).

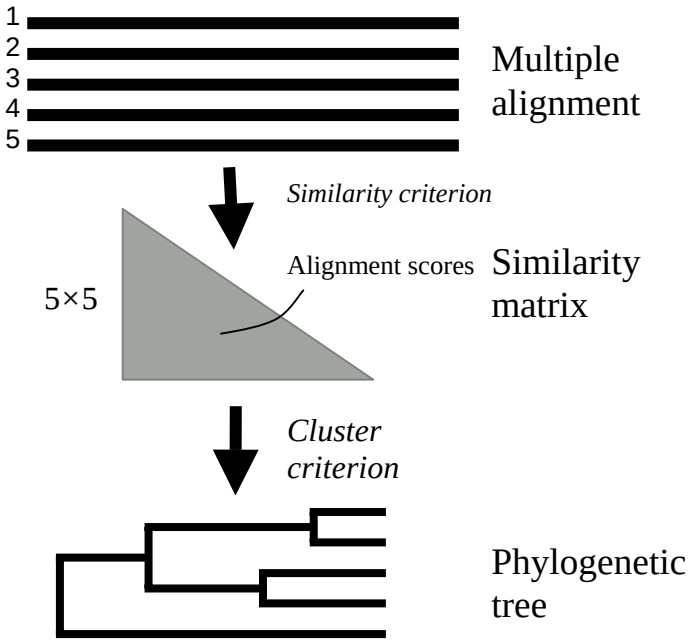


Figure 2.1. A MSA gives rise to a similarity matrix containing all pair-wise distances, which can be clustered and represented by a phylogenetic guide tree.

2.2. REPRESENTING SEQUENCE AND SEQUENCE BLOCK INFORMATION

2.2.1 Trees

Reconstruction of the evolutionary history of proteins is one of the central aims of sequence comparison. An evolutionary model of a particular sequence family consists of an evolutionary tree depicting the sequence relationships within the family and a MSA (Figure 2.1), which shows the detailed local relation between the individual sequences. We adopt the following conventions and terminology: a) trees consist of edges (lines) and nodes (crossings), where edge lengths define the distance between sequences and nodes the actual sequences; b) trees are binary, with one incoming edge and two outgoing edges; c) the ultimate ancestor is called 'root', the terminal nodes are called 'leaves'. Some tree construction algorithms (like parsimony) give no indication about the position of the root, leading to 'unrooted' trees. For the sake of completeness (and some erroneous formulae in the standard literature), we give here the equations for the number of possible unrooted and rooted trees for *n* sequences (Figure 2.2).

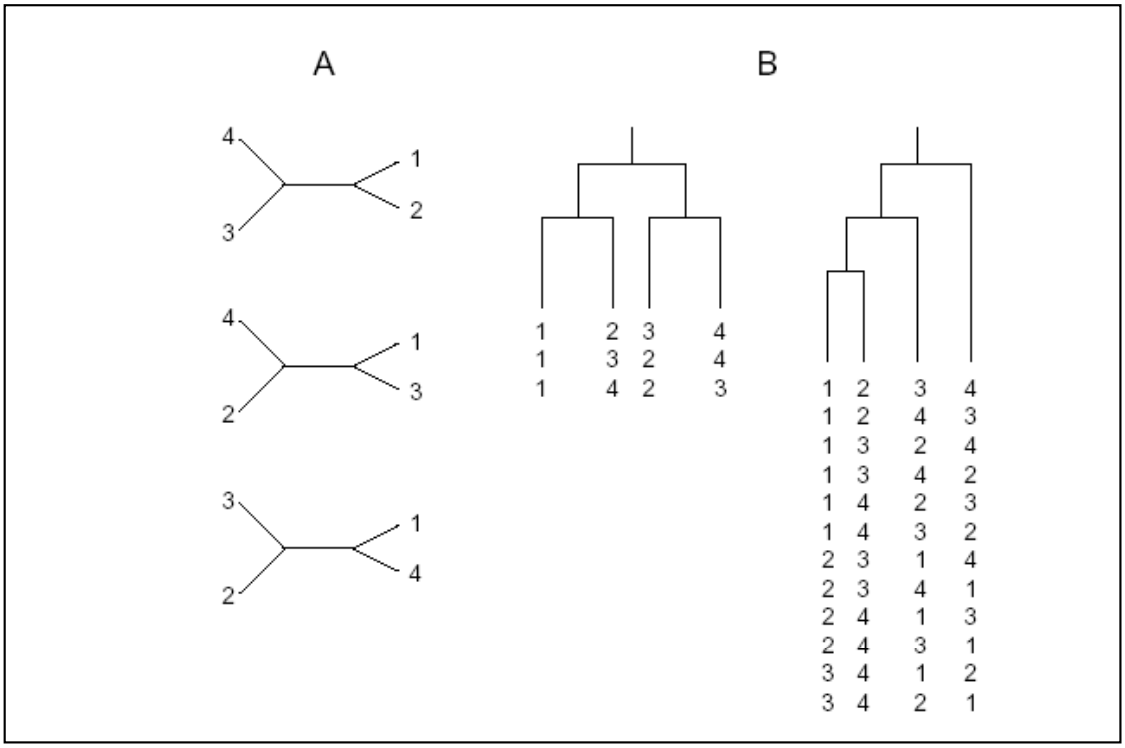


Figure 2.2. Illustration of all phylogenetic trees for a set of four sequences. A) There are three different possible unrooted trees; B) Two different tree topologies and a total of 15 different rooted trees.

$$\textit{Number of unrooted trees} = \frac{(2n-5)!}{2^{n-3}(n-3)!} \quad (1)$$

$$\textit{Number of rooted trees} = \frac{(2n-3)!}{2^{n-2}(n-2)!} \quad (2)$$

Rooting of a tree can be achieved by adding a very distant member, also called an outlier group, to the family, which defines the root as the point where its branch meets the tree. The construction of 'real' phylogenetic trees is a computer-intensive procedure that requires probabilistic modelling. A widely used approach is the application of maximum likelihood methods, which calculate the most probable phylogenetic tree associated with a give MSA and evolutionary model (Saitou, 1990). For the purpose of constructing a guide tree for a MSA, more amenable *ad hoc* strategies are often adopted to reduce the computations. A frequently used approach is to estimate sequence distances from the pair-wise alignment scores. Using these heuristic distances, a phylogenetic tree can be constructed to guide the construction of the MSA (see section “Progressive Alignment Strategies”).

Regardless of whether the phylogenetic tree is calculated using maximum likelihood or distance methods, the significance of the branching of the tree can be estimated using the bootstrap (Felsenstein, 1985). For bootstrapping, the columns in the multiple alignment are re-sampled a significant number of times (100-1000) with replacement, such that a single alignment column can occur multiple times in a re-sampled MSA, after which the significance for each branching in the original tree is taken from the frequency of the occurrence of the branch over all re-sampled trees (Figure 2.3).

The transformation from pair-wise alignments to a phylogenetic tree is performed by clustering algorithms, which fall into two conceptually different categories: distance-based (UPGMA, Neighbour-Joining) and parsimony. Distances can be obtained from sequence identities (Fitch and Margoliash, 1967) or pairwise sequence alignment scores (Hogeweg and Hesper, 1984).

UPGMA (Un-weighted Pair Group Method using arithmetic Averages) (Sokal and Michener, 1958) joins sequences to clusters and progressively to larger clusters until a single cluster (tree) is accomplished. The order by which sequences or clusters

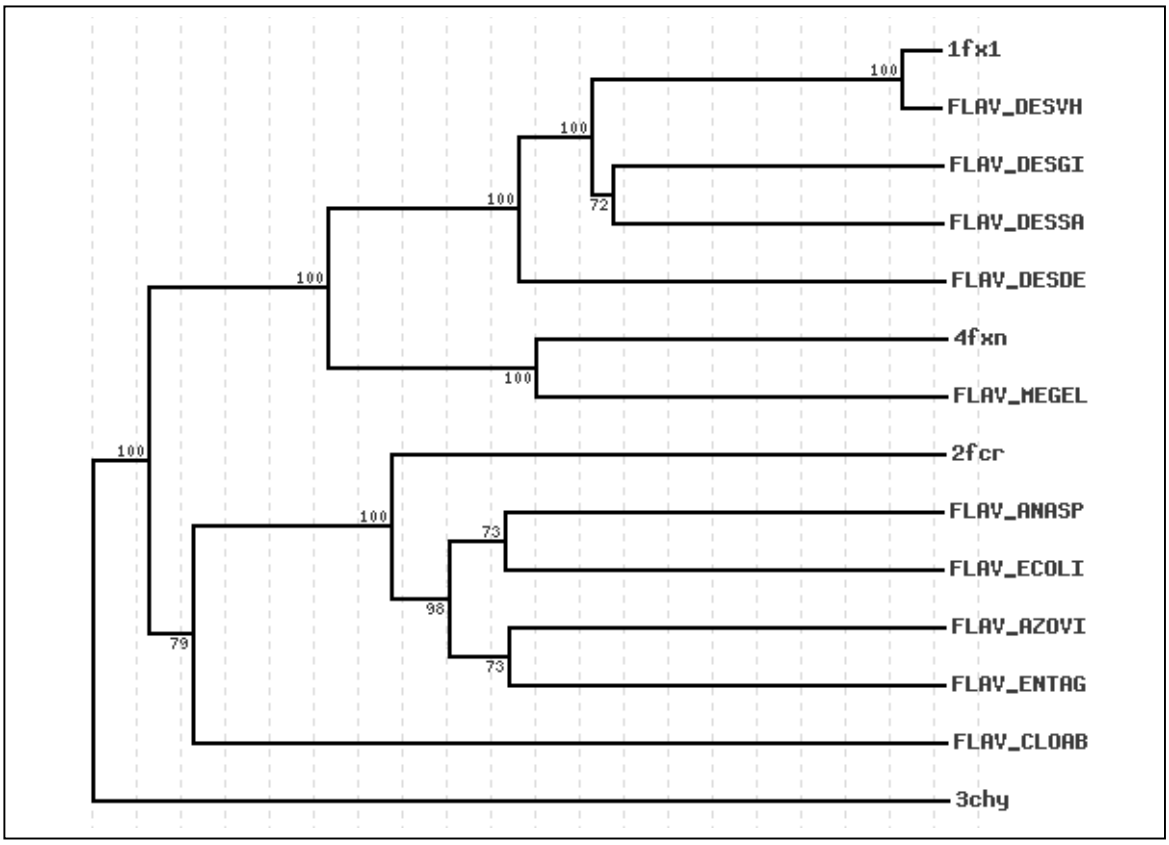


Figure 2.3. The phylogenetic tree of the flavodoxin family. The numbers at the ancestral nodes are bootstrap values.

are joined is simple: always the closest pair of sequence/sequence, sequence/cluster or cluster/cluster is joined. Each joining operation creates an ancestral node, and the distance between the joined sequences (which are the leafs of the tree) is expressed as

$$\textit{Distance between joined sequences} = \frac{1}{n_i \cdot n_j} \sum d_{ij} \qquad (3)$$

where n_i, n_j are the number of sequences in the two joined clusters, and d_{ij} are the distances of all possible pair-wise sequence combinations between the joined clusters. For example, if cluster 'A' consists of sequences (1,2,3) and cluster 'B' consists of sequences (4,5), the distance is $\frac{1}{3 \cdot 2}\left(d_{14} + d_{15} + d_{24} + d_{25} + d_{34} + d_{35}\right)$. The ancestral node is exactly in the middle between the joined sequences, so that the edge length between the joined sequences and their ancestral node is 1/2 of the above distance.

Neighbour-Joining (NJ) (Saitou and Nei, 1987) should be performed instead of UPGMA if the distances between sequences are not additive, which is equivalent with

an un-scaled evolutionary clock resulting from unequal evolutionary speeds in the various branches. The algorithm starts from a distance table containing all pair-wise sequence distances by joining the closest pair of sequences and placing their ancestral node at half distance. The distance table is updated: the two columns/rows of the joined sequences are fused together and the average distance of the two joined sequences to every other sequence is computed. For example, for a sequence set A, B, C, D, after joining the closest sequence pair A and B to (AB), the average distances of (AB) to C and D are $\frac{1}{2}(\mathrm{AC}+\mathrm{BC})$ and $\frac{1}{2}(\mathrm{AD}+\mathrm{BD})$. The closest pair, assuming (AB) and C, is joined to (ABC) and their ancestral node is placed at half distance. The procedure of average distance computation and joining is repeated until all sequences are included. The resulting tree is unrooted.

The Maximum Parsimony (MP) method is an algorithm to find the tree that minimizes residue substitutions summed up over all sites of the whole tree (Eck and Dayhoff, 1966; Kluge and Farris, 1969). Therefore, a sufficient number of different tree topologies is generated in a first phase, and a cost for residue substitutions is assigned to each tree in a second phase. This cost function can be simply the total number of residue substitutions or the sum of weights W_{AB} for each substitution from residue A to residue B (weighted parsimony). The algorithm of Fitch (Fitch, 1971) is usually used for counting the number of residue changes. Sequences on ancestral nodes can be inferred if pointers between residues on ancestral and daughter nodes are stored. The number of possible tree topologies increases drastically with the number of sequences (*vide supra*), but stochastic approaches have been developed (Felsenstein, 1981).

2.2.2 Profiles

A MSA profile is a comprehensive representation of a MSA, stressing the composition of the alignment positions (columns) rather than the composition of the constituting sequences. In general form, a profile is a vector composed of 20 components (amino acids) at each MSA position. The vector components describe the contribution (score, weight, probability etc.) of each amino acid at this position to the MSA.

It must be stressed however, that in the context of progressive multiple sequence alignment the application of pseudo-counts, and thus incorporating background amino acids frequencies, can well decrease proper alignment, notably during early steps of progressive alignment when sequence blocks to be aligned only contain a single or few sequences (Heringa, personal communication). Strict Bayesian modelling treats model parameters for prior information as distributions rather than single values. Such distributions can be described by Dirichlet densities or mixtures. A Dirichlet mixture is a probability density over a set of probability vectors, in our case vectors containing the probabilities of 20 amino acids as components, so that each vector describes a different probability distribution of the amino acids (Sjolander et al., 1996).

Another flexible motif search technique introduced by Bucher *et al.* (Bucher et al., 1996) uses ‘generalised profiles’, which are similar to HMMs. A profile is represented by the sequence alphabet and the possible states of an alignment that are defined as *begin, match, insert, deletion* and *end*.

2.2.3 Consensus

A consensus sequence represents the most reduced form of a profile, with each position having one component set to one (the consensus amino acid) and all others to zero. A straightforward way to construct a consensus sequence is to choose the most frequent or most likely residue at each alignment position. Although appealing because of its simplicity, a consensus sequence carries less information than a MSA (thus it is a degenerated representation), which may lead to misinterpretations in comparisons of the consensus sequence with related sequences (Schneider, 2002). A related but more sensitive way to compress the consensus information given by a MSA is through 'partial order graphs' (Lee et al., 2002), which can be viewed as a formalism that provides multiple alternative consensus sequences for non-conserved MSA regions. A partial order graph of similar sequences contains a main 'consensus' branch for conserved MSA segments and loops where sequences diverge from each other. Despite this condensed representation, the entire information of the MSA is retained. A new sequence to be aligned against the MSA will then be aligned to such non-conserved regions through the most similar sequence within the alignment.

2.3. PERFORMING MULTIPLE SEQUENCE ALIGNMENT

Carrying out a MSA of a given protein sequence set and extracting maximum information from the alignment involves a number of critical steps:

- The selection of sequences
- The choice of the scoring function used to compare sequences or sequence blocks
- The application and optimisation of this scoring function in compiling the alignment

2.3.1 Selecting the sequences for an MSA

A MSA can be misleading when a sequence set contains sequences that are not homologous. Ideally, the sequences should all be orthologues, but in practise it is often difficult to ensure that this is the case. It should be stressed that most MSA routines will produce an alignment even in the case of biologically unrelated sequences, which can give rise to spurious suggestions regarding the proteins' structure or function ('garbage in garbage out'). A widely used way to create a sequence set around a given query sequence of interest is to employ a homology searching technique (Altschul et al., 1990; Altschul et al., 1997; Altschul and Koonin, 1998; Eddy, 1998; Karplus et al., 1998; Taylor, 1998; Taylor and Brown, 1999; Karplus et al., 2001; Yona and Levitt, 2002; Ginalski et al., 2003; Sadreyev et al., 2003; Capriotti et al., 2004; Edgar and Sjolander, 2004a; Ginalski et al., 2004; Soding, 2004) to scour sequences in public sequence databases. Although the development of P- and E-values to estimate the statistical significance of putative homologues found by these programs limits the chance of false positives, it is entirely possible that essentially non-homologous sequences enter the alignment set, which might confuse the alignment method used.

2.3.2 The MSA scoring function

The scoring function is the formalization of the biological knowledge used in aligning the sequences. Ideally it should contain all available knowledge about evolutionary, structural and functional aspects of the compared sequences, so that the

scoring function approximates the biological reality. In practice, however, this information is often not available or cannot be formalized mathematically.

a. Scoring single-to-single sequence comparisons

Although each cross-comparison of a residue between two sequences should in reality be evaluated individually based on its structural and functional context, the most widely used scheme to compare sequences is based on generalized averages for scoring each pair of residue types, given in the form of a symmetric 20×20 amino acid exchange matrix. The scheme models the alignment of two sequences as a Markov process, where the amino acid matches are considered independent, so that the product of the probabilities for each match within an alignment can be taken. Since many of the generally applied 20×20 scoring matrices contain propensities converted to logarithmic values (*log-odds*), the alignment score S of two single sequences can normally be calculated by summing the log-odd values corresponding to matched residues minus appropriate gap penalties:

$$S = \sum_l s(a,b) - \sum_k N_k \cdot gp(k) \tag{4}$$

where the first summation is over the exchange values $s(a,b)$ associated with l matched residues and the second over each group of gaps of length k, with N_k being the number of gaps of length k and $gp(k)$ the associated gap penalty.

In case affine gap penalties are used, $gp(k) = pi + k \cdot pe$, where pi and pe are the penalties for gap initialisation and extension, respectively. A consequence of the widely used affine gap penalty scheme is that long gaps required, for example, to span a domain *B* in aligning a two-domain sequence *AC* (where *A* and *C* represent domains) with a three-domain sequence *ABC*, are often too costly, so that such sequences become misaligned. A complication with gap penalties is that there exists no formal model to set their values according to the evolutionary distances suggested by the exchange values within scoring matrices, so that one has to resort to empirical tuning of the gap penalties.

b. Scoring multiple-to-single and multiple-to-multiple sequence comparisons

In order to align a MSA to a single sequence or to another MSA, profiles are used. Scoring of such multiple-to-single or multiple-to-multiple is in a theoretical sense far from trivial, because the substitution matrices are strictly derived from probabilities of pair-wise residue alignments. However, classical pair-wise substitution scores are commonly averaged or condensed by some other linear transformation over the representing amino acids at each alignment position, yielding a 'position-specific scoring matrix' (PSSMs).

The equation for the average profile score *S* of two profile alignment positions (columns) *x* and *y* reads:

$$S_{xy} = \sum_{i=1}^{20} \sum_{j=1}^{20} f_i f_j S(i, j) \tag{5}$$

where *i* and *j* denote the amino acids represented in each profile, f_i and f_j are the frequencies of amino acids *i* and *j* in alignment position *x* and *S(i, j)* is the substitution score of amino acids *i* and *j* (Gribskov et al., 1987). The average profile score is appropriate for alignments with a large number of sequences (*N*), but gives poor results for small *N* when *f/N* deviates from the expected probability *p* to find a residue at position *x*. Therefore, it is advantageous to add a quantity proportional to the background probability of each amino acid to the real frequency *f*, yielding:

$$S_{xy} = \sum_{i=1}^{20} \sum_{j=1}^{20} (f_i + Aq_i)(f_j + Aq_j) S(i, j) \tag{6}$$

where the term *Aq* is called 'pseudo-counts', with constant *A* as a weight of the pseudo-counts (relative to *f*) and *q* as background frequency of the corresponding residue. There is a good theoretical justification for the use of pseudo-counts within the framework of Bayesian statistics, where they represent the prior information about the data (Durbin, 1998).

In the event that a profile is compared (aligned) to a single sequence, equations 5 and 6 can be simplified since the *j* component is no longer a profile column but a single residue and therefore its frequency f_j is 1 and is always the same for all *S(i,j)* exchange weights:

$$S_{xy} = \sum_{i=1}^{20} f_i S(i, j) \tag{7}$$

$$S_{xy} = \sum_{i=1}^{20} (f_i + Aq_i) S(i, j) \quad (8)$$

Regardless of whether an alignment is between two profiles or a profile and a sequence, the above compression of the information in a profile alignment position allows the remodelling of equation 4 to calculate the score *S* of a profile-sequence or profile-profile alignment by:

$$S = \sum_l S_{xy} - \sum_k N_k \cdot gp(k) \quad (9)$$

where the first summation is over *l* matched average profile position (column) scores s_{xy} and the second over each group of gaps of length *k*, with N_k being the number of gaps of length *k* and *gp(k)* the associated gap penalty. Affine gap penalties can also be used in these cases, although many profiles also include position-specific gap penalty adjustments based on the alignment information.

More recently, an alternative profile scoring function has been proposed, known as 'log-average scoring' (von Ohsen and Zimmer, 2001; von Ohsen et al., 2003). The difference to the original scoring function is simply that instead of summing the log-odds values of propensities stored in the 20×20 scoring matrices, the original propensities are first added and then the log of the sum is taken. The result of this simple switch is that in the comparison of two sequences or profiles, the 'log-average' alignment score now represents the global probability that these two sequences or profiles are related (based on the evolutionary model of the scoring matrix used), rather than the cumulative independent probabilities of the matched amino acid pairs. The 'log-average' score has been implemented in the recent progressive multiple alignment method MUSCLE (Edgar, 2004) and in addition has been further customised for the needs of multiple sequence alignment by the addition of position-specific gap penalties and renamed to 'log-expectation' scoring (Edgar, 2004).

2.3.3 Applying the scoring function

Apart from being a fundamental biological challenge, MSA is also a computationally intense problem. The closest to an exact solution are algorithms that perform simultaneous alignment over a multidimensional search matrix, where each

sequence in the MSA represents an extra dimension (Lipman et al., 1989; Stoye et al., 1997; Stoye, 1998).

The most populated class of algorithms is that of progressive MSA methods. The progressive strategy implies that an algorithm for pair-wise sequence alignment is repeatedly used in a step-wise fashion until all sequences are aligned (Feng and Doolittle, 1987). In the vast majority of progressive methods the Dynamic Programming (DP) strategy is adopted. The DP computational technique was originally developed by the renowned mathematician Dr. Richard Bellman in 1953 and was introduced to biological sequence alignment research in 1970 by Needleman and Wunsch (Needleman and Wunsch, 1970). The DP strategy guarantees that, given an amino acid exchange matrix and gap penalty values, the highest scoring or optimal pair-wise alignment is calculated. The progressive alignment strategy reuses the pair-wise DP algorithm in a greedy manner; *i.e.*, alignments formed during the progression towards the final MSA cannot be changed anymore. The main difference between the available DP-based methods is the way in which the information of aligned blocks of sequences is represented (see section “Profiles”). While early methods used consensus sequences to represent alignment blocks, current methods all use a profile formalism to represent the information in a MSA (Gribskov et al., 1987). Recent developments in multiple alignment techniques have mainly focused on sensitive and optimal models to represent MSA information.

A class of techniques that are able to revisit and optimise is that of iterative multiple alignment techniques. Pioneered by Hogeweg and Hesper (Hogeweg and Hesper, 1984), iterative techniques attempt to enhance the alignment quality by gleaning information from a multiple alignment constructed in an earlier round, which is then applied in a next round to improve the alignment according to a given scoring scheme. Other classes of alignments include stochastic alignments, where probabilistic frameworks such as Hidden Markov Models and Bayesian networks have been attempted, as well as fast computational techniques such as suffix trees and fast Fourier transforms (FFT).

In the remainder of this chapter, general methodological issues will be covered in the next section, after which an overview of current state-of-the-art methods will be

presented. Finally, the last section offers some considerations on benchmarking and training issues.

2.4. MSA METHODOLOGY

In this section we focus on higher-level aspects of modelling, concerning the complex relationships within sequence families, such as tree construction and progressive alignment strategies. These strategies are discussed along with practical considerations about the construction of meaningful MSAs.

2.4.1 Progressive Alignment Strategies

To increase the chance of correct alignment, many methods calculate an appropriate order in which the sequences will be aligned progressively (Figure 2.4). In many cases, this order is derived from all-against-all pair-wise alignment of the sequences and the calculation of a dendrogram, often referred to as the "guide tree", using the pair-wise alignment scores (see section "Trees"). The resulting branch order of the dendrogram is then followed to align the sequences, such that the most similar sequences are aligned first, and gradually the more distant sequences are included in the growing MSA. Although an efficient and stable strategy, the progressive alignment protocol suffers from its greediness and is not able to revise any of the alignments made earlier, such that any alignment errors during the construction of the MSA cannot be repaired anymore. This drawback of the strategy is particularly significant for distant sequence sets because the comparisons of sequences at early steps during progressive alignments cannot make use of information from other sequences, so that proper positional information required for correct matching is not available at early stages. It is only later during the alignment progression that more information from other sequences (e.g. through profile representation) becomes employed in the alignment steps, but quite possibly after misalignment has already taken place.

MSA programs like ClustalW (Thompson et al., 1994), T-COFFEE (Notredame et al., 2000), PRALINE (Heringa, 1999, 2002) and MUSCLE (Edgar, 2004) are based upon the progressive alignment strategy (Feng and Doolittle, 1987) and are all able to produce high-quality alignments as demonstrated in a recent benchmark (Heringa, 2002) over 144 alignments in the BAliBASE repository (Thompson et al., 1999a),

although their results are not necessarily identical, particularly with more divergent sequence sets.

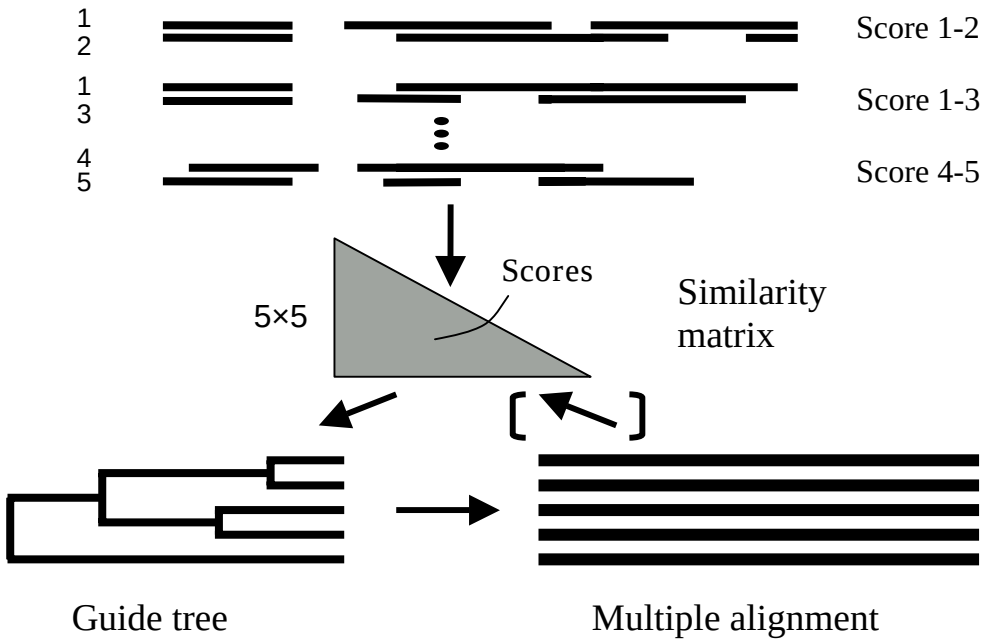


Figure 2.4. Representation of the progressive alignment strategy comprising compilation and scoring of all pairwise alignments, yielding a similarity matrix, which is used to construct a guide tree. The resulting MSA is constructed in the order as given by the guide tree. The arrow in brackets represents alignment iteration.

2.4.2 Positional conservation

Positional conservation is an important measure for detecting homology. Conserved alignment blocks are often described as 'motifs', implicating structurally and/or functionally important parts of proteins. Frequently even highly divergent sequence families share common motifs; sometimes such motifs are the only indication for sequence relatedness. Some databases are derived from grouping sequences with common alignment blocks or motifs into families (BLOCKS, FSSP).

A problematic aspect is the relation between positional conservation and sequence conservation. When sequences of high pair-wise sequence identity are aligned, positional conservation scores are accordingly high, but mostly due to redundancy rather than true evolutionary conservation. In other words, sequences that are close in evolutionary time yield little information about true conservation patterns.

This consideration has led to the usage of various weighting schemes: tree-based (Altschul, 1989; Thompson et al., 1994), pair-wise distance-based (Vingron and Argos, 1989; Sibbald and Argos, 1990; Vingron and Sibbald, 1993) and position-based (Henikoff and Henikoff, 1994).

2.4.3 Simultaneous alignment

The dynamic programming algorithm for pair-wise sequence alignment can be extended to multiple sequences (Murata et al., 1985; Gotoh, 1986) by using a multi-dimensional search matrix. However, the dimensionality of the search matrix is equal to the number of sequences and the search space equals the product of all sequence lengths ($O(L^N)$), where L is the average sequence length and N the number of sequences), rendering the search unfeasible even for moderately sized alignments. Approaches to reduce the computational load comprise reduction of the search space to near-diagonal paths (Carillo and Lipman, 1988; Wang and Jiang, 1994; Stoye et al., 1997), pre-selecting similar segments (Johnson and Doolittle, 1986), or word matching (Sobel and Martinez, 1986; Waterman, 1986; Vingron and Argos, 1989; Waterman and Jones, 1990).

A more recent development is mimicking simultaneous alignment by using pre-compiled profiles or libraries in progressive alignment. For each sequence, such a library contains information from other sequences and thus extends the sequence information used for each pair-wise alignment. The idea behind multi-sequence profiles or libraries is to accumulate information from global and local alignments as well as from various sequence groupings. The accumulated information for each sequence is considered to be more reliable than single sequence alignments alone, so that match errors during progressive alignment are reduced.

The information for each sequence can be gathered by using pair-wise alignments to construct a master-slave alignment. In the program PRALINE (Heringa, 1999, 2002), *N* master-slave alignments are constructed for *N* sequences, where each sequence in turn is the master sequence. The inclusion of slave sequences can be adjusted by a score threshold, so that sequences deemed too divergent are excluded and perturbation of the conservation pattern is avoided. The master-slave alignments are then converted into pre-profiles and used for the progressive construction of a final

alignment. The advantage of this method is the possibility to combine local and global alignment information. Moreover, sequences contained in multiple pre-profiles can be used to derive position-specific consistency scores, which effectively measure the agreement between the multiple alignment and pair-wise alignments.

A combination of local and global alignment is also achieved by the program T-COFFEE (Notredame et al., 2000). Information from local and global pair-wise alignments is complemented with information from triplet alignments that provide an alignment for each considered pair of sequences through each possible third sequence. For each pair of sequences, the contributions from the direct pair-wise alignment and the triplet alignments are combined in a position-specific weight library (library extension), yielding a weight for each aligned residue pair. This library is then used to construct a final alignment by dynamic programming following the progressive alignment strategy.

2.4.4 Alignment Iteration

As mentioned in the above sections, the central aim of MSA methodology is to capture the complex evolutionary relationship between sequences and to convert the biological reality into a sensible scoring scheme. The ultimate step is to find the optimal mutual sequence arrangement by maximising the alignment score. While strict (multi-dimensional) dynamic programming guarantees to find the optimal (multiple) alignment score, heuristic procedures such as progressive alignment do not necessarily yield the optimal score.

A class of techniques able to revisit and optimise a MSA is that of iterative multiple alignment techniques). Pioneered by Hogeweg and Hesper (Hogeweg and Hesper, 1984), iterative techniques attempt to enhance the alignment quality by gleaning information from a multiple alignment constructed in an earlier round, which is then applied in a next round to improve the alignment according to a given scoring scheme. Iteration can be employed to further increase the alignment score, the incentive being to reach the optimum by 'hill climbing', *i.e.* stepwise increase of the target function (alignment score) until convergence is reached. During iteration, the order at which the sequences are progressively aligned can be altered (Hogeweg and Hesper, 1984; Gotoh, 1996) or other criteria derived from a multiple alignment produced in the

preceding round can be applied as an iterative scoring scheme (Heringa, 2002). This means that the target function of the iteration process can be different from the alignment score. In some cases it is desirable to maximise consistency, conservation or some other function specific to the alignment problem.

Iteration is a reasonably efficient and robust technique that alleviates the greediness of the progressive strategy. Results are critically dependent on the scoring scheme used and often there is no certainty that convergence will be reached, and if so, whether the converged multiple alignment is biological more optimal than earlier ones. The other two possible scenarios in addition to convergence are divergence, in which the program enters a route through a virtually infinite number of states, and limit cycle, in which the program visits recursively a finite number of states. In cases where a different target function that the alignment score has been used to guide iteration, a decision has to be made whether the last (with the maximal target function value) or the highest scoring alignment will be taken as the result after convergence has been reached. A choice between several solutions also exists in the outcome of the limit cycle and divergence scenarios. It is the task of the investigator to perform alignment iteration with intuition and knowledge, in order to choose the right combination of target functions, alignment strategies and iterations to gain information about the sequence set under consideration.

A recent investigation of different iteration strategies (Wallace et al., 2004) has shown that the use of iterative optimisation has beneficial effects even on the most recent state-of-the-art multiple alignment programs, such as T-COFFEE (Notredame et al., 2000), MUSCLE (Edgar, 2004) and ProbCons (Do et al., 2005). The same study has also provided a possible solution to the limitations observed by Ebedes and Datta (Ebedes and Datta, 2004) in the parallelisation of ClustalW (Thompson et al., 1994), the most widely used multiple alignment method in biological research.

2.4.5 Probabilistic MSA

Generation of a MSA can be performed entirely in a probabilistic framework. The probability of observing a certain sequence can be inferred using a Hidden Markov Model (HMM). An HMM consists of character-emitting states and transitions between these states. Character emissions and state transitions are connected to probabilities (the

probabilistic model) that determine the behaviour of the HMM. To create a sequence, an HMM generates a Markovian path through states, i.e. each step, including character emission and transition, is independent of the previous step. Given a sequence, the probability of observing this specific sequence can be derived using the Viterbi algorithm (Viterbi, 1967). Pair-wise alignment algorithms for HMMs have been described (Durbin, 1998). For progressive MSA, the pair-wise approach is extended to include the phylogenetic inter-dependence of sequences. Probabilistic modelling of sequence alignments is theoretically and computationally involved, and was in the past largely restricted to specialists. However, recent improvements in program development and computer speed have made this approach more accessible and the quality of probabilistic alignments for nucleotide sequences is now comparable to that of standard alignment methods.

Another probabilistic modelling approach is Bayesian inference (Liu and Lawrence, 1999), where all unknown variables are treated as probability distributions. The advantage of Bayesian modelling is that it allows for inclusion of prior knowledge about the system. However, the computational burden can be prohibitive, and specific sampling techniques such as 'general Markov chain Monte Carlo' may be required (Tanner and Wong, 1987). A collection of algorithms for Bayesian alignment has been described by Zhu *et al*. (Zhu et al., 1998).

2.4.6 Parallelisation of MSA

With the increasing availability of computer clusters in computationally oriented labs, it is worth considering parallelisation of the most time-consuming computational tasks. Highly repetitive procedures, such as the pair-wise sequence or profile alignment phase in progressive alignment, are favourable targets for parallelised (or distributed) computing. Parallelised programs are designed to split the total computational task into sub-tasks that are processed on separate CPUs (nodes). Implementation of parallelised code typically requires to identify the most CPU-intensive task (frequently a loop structure) and to split it into sub-tasks for independent execution. For example, a MSA of 4 sequences requires 6 pair-wise alignments (sub-tasks), which can be performed, for example, in blocks of 3 on 2 nodes (CPUs), or in blocks of 2 on 3 nodes.

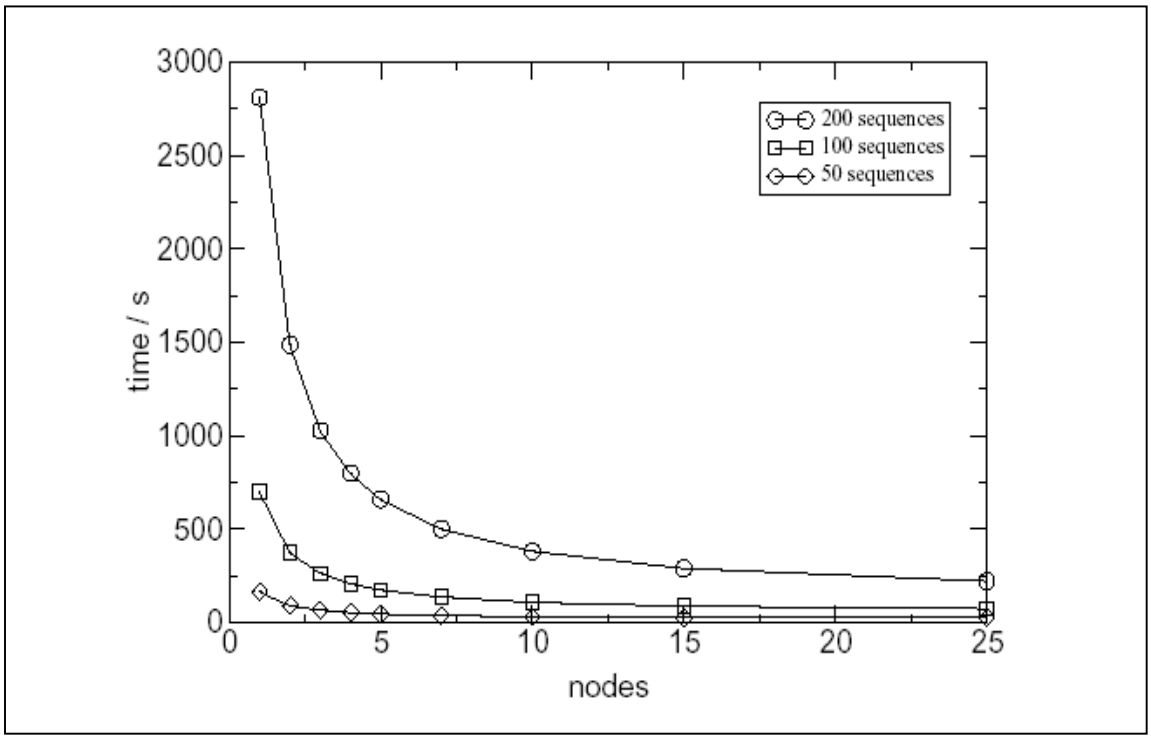


Figure 2.5. Computational times of parallelised PRALINE on different numbers of nodes for three sets of 200, 100 and 50 sequences, each 200 residues long (Kleinjung et al., 2002).

The technical details of parallelisation are dealt with by high-level routines provided by a parallelisation interface such as the 'message passing interface' package MPICH, available at http:/www-unix.mcs.anl.gov/mpi/mpich (Gropp et al., 1996; Pacheco, 1997). If all nodes execute the same operations but perform these on different sub-sets of distributed data, the parallelisation technology is called single-instruction multiple-data (SIMD). Parallel code is most efficient at a minimum amount of communication between the nodes and at optimal balancing of the computational load over the CPUs.

The MSA program PRALINE (Heringa, 1999, 2002) has been parallelised in the form of SIMD technology (Kleinjung et al., 2002). The scaling of computational times *versus* the number of employed nodes is plotted in Figure 2.5 for three differently sized sets of sequences. Parallelised PRALINE generated a MSA up to ten times faster than the single processor version, when tested on a set of 200 random sequences of 200 residues length.

2.5. THE ORIGINAL PRALINE METHOD

PRALINE (PRofile ALIgNmEnt) (Heringa, 1999) is a global progressive alignment algorithm that re-evaluates at each alignment step which sequences or blocks of sequences should be aligned and hence determines the order in which sequences should be aligned on the fly. The pair-wise alignments are performed using dynamic programming.

PRALINE is a MSA toolkit that integrates a number of strategies for the optimisation of MSA quality, such as local global alignment, global and local profile pre-processing, secondary structure-guided alignment and has weighted iteration capabilities. The unique feature of PRALINE is that it allows the combination of its different optimisation strategies, rather than limiting the creation of a MSA to a single approach. Consequently, matching the best alignment strategy to the problem at hand can ensure a high quality MSA. PRALINE has also been parallelised (Kleinjung et al., 2002) yielding a tenfold acceleration compared to single processor execution (see section “Parallelisation of MSA”). The PRALINE alignment optimisation strategies are described below:

The *local global optimisation* strategy is aimed to identify regions of useful local information and use it to guide the final global alignment (Heringa, 1999, 2002).

The *profile pre-processing* strategy can be applied to both global and local alignment strategies (Heringa, 1999, 2002). The principle is that each sequence is represented as a pre-profile that contains the position specific information derived from the initial all-against-all pair-wise alignments it is involved in. In addition, these pre-profiles can be processed for each sequence by only including information from those pair-wise alignments that score beyond a user-specified threshold. A low threshold would result in a pre-profile for each sequence comprising information from all other sequences, while higher thresholds would tend to incorporate only increasingly related information. Once all pre-profiles are created, the progressive alignment strategy proceeds as described above using the pre-profiles instead of single sequences. The benefit of this is that the information from other sequences (in particular similar sequences) and the use of position-specific gap penalties ensure that gaps are not inserted in un-gapped (core) regions and also that during the progressive strategy distant sequences are no longer considered independently at the last alignment steps.

The MSAs of the profile pre-processing strategy can be further optimised in a *weighted iterative scenario*. In the profile pre-processing strategy, each sequence in the final alignment can be assessed in terms of the degree of consistency aligned residue pairs have reached across the pre-profiles (residue pairs that are consistently aligned are more trustworthy than others that vary between pre-profiles). This consistency information is stored and when the MSA is iterated, the succeeding round uses this information as weights in the dynamic programming to optimise the final alignment (Heringa, 2002). From the resulting set of iterative alignments, the one with the highest cumulative score over all pair-wise matched amino acids in the alignment (sum-of-pairs score) can be selected as a safeguard to prevent alignments from wandering away to less optimal areas in the alignment space (Heringa, 2002).

2.6. OTHER STATE-OF-THE-ART MSA METHODS

MSA is an intricate problem and over the past 30 years an increasing number of approaches have been developed that try to solve it, each with its own strengths and weaknesses. Unfortunately, the diversity of the methodologies makes it difficult for non-specialists to know which method is the best to use for a particular problem. A sensible decision can be made with a clear and thorough understanding of how the methods work, where they perform well and what their limitations are. The methods described below are an assembly of the most commonly used top-performing methods to date and useful guidelines are proposed based on published assessments, where available.

2.6.1 CLUSTALW, CLUSTALX

ClustalW (Thompson et al., 1994) and the later window graphic user interface (GUI) version ClustalX (Thompson et al., 1997) are the newest versions of the global progressive alignment algorithm Clustal (Higgins and Sharp, 1988) and are generally considered as the standard method for MSA. The progressive strategy used is a simplification of the original Feng and Doolittle scheme (Feng and Doolittle, 1987). The alignment is constructed by first building a guide dendrogram using Neighbour-Joining (NJ) (previous versions used the UPGMA strategy), based on sequence similarity, which is subsequently used to order successive pair-wise alignments. The

already aligned sequences are reduced to a profile for the subsequent pair-wise alignment (previous versions used position consistencies). However, during the progressive alignment process, highly specialised heuristics are applied to try and optimise how the sequence information is processed: When the sequences are ordered for alignment according to the pre-computed dendrogram, the alignment of distantly related sequences is delayed, thus overriding the dendrogram. This is implemented to correct for the limitation of progressive alignment, which does not allow alterations of the alignment once a sequence has been aligned even if later added sequences may require it, “once a gap always a gap” (Feng and Doolittle, 1987). Also the pair-wise alignments are performed using local gap penalties and there is automatic selection and adjustment of the residue substitution matrix and gap penalties, respectively.

The algorithm is reasonably fast and can handle large sets of sequences, but speed becomes an issue when it is given genomic compared to other available methods (Lassmann and Sonnhammer, 2002) such as POA. Possible insights for its parallelisation have been recently described (Ebedes and Datta, 2004; Wallace et al., 2004).

2.6.2 POA

The Partial Order Alignment (POA) (Lee et al., 2002) is an extension of the conventional dynamic programming approach. Instead of performing pair-wise alignments following a specific order (from a guide tree), sequences are aligned in the order in which they are given. The growing MSA is represented by a 'partial order graph', in which identical residues within a column are fused and the information of the sequence origin is stored. Thus, despite the condensed representation, the entire information of the MSA is retained. A typical PO-MSA of similar sequences contains a main 'consensus' branch and loops where sequences diverge from each other. The POA dynamic programming matrix reflects this structure by adopting the bifurcation points, so that the matrix consists of multiple two-dimensional layers that part and re-join according to the PO-MSA graph. The best alignment is found by a conventional trace-back operation. The POA algorithm guarantees that each sequence is aligned to the closest sequence in the growing MSA.

A

… PLIVQNEPDEAEQDC …
… AI-VNEMPDNA-PEC …

B

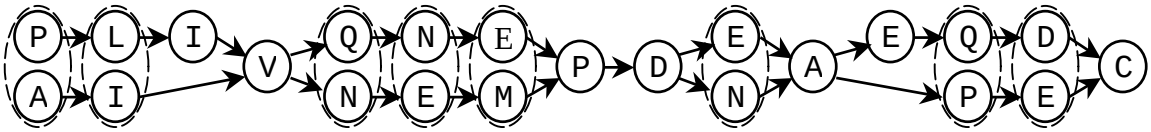


Figure 2.6. The Partial Order Graph (POA) alignment representation of the C-termini of a pair of flavodoxin proteins. A) The alignment in standard format; B) The alignment in POA representation (adapted from (Lee et al., 2002)).

POA (Partial Order Alignment) is a novel local progressive algorithm. The novel feature of this method is that it employs partially ordered graphs to represent aligned sequences instead of profiles (see section “Profiles”). The POA dynamic programming matrix reflects this structure by adopting the bifurcation points, so that the matrix consists of multiple two-dimensional layers that part and re-join according to the PO-MSA graph. The progressive strategy for this method does not follow a guide dendrogram to determine the order in which the sequences will be aligned, but aligns the input sequences in the order in which they are given. Each time a new sequence is added to the growing alignment, it is aligned with the most closely related hybrid sequence within the MSA as given by the partial order graph. Pair-wise alignments are performed using the Smith-Waterman algorithm (Smith and Waterman, 1981), which is extended to accommodate the partial order graph representation. The partial order graph is constructed in two main steps: first, the sequences are converted to PO-MSA (Partial Order-MSA) data structures. Then, the closest related pair of PO-MSAs is aligned and the identical residues are fused into nodes (like the knots on two ropes tied together at points along their length), while the remaining residue origins and positions are recorded and considered as incoming and outgoing (directed) edges from each node (the rope “bubble” before and after each knot) (see Figure 2.6). When the partial order graph is then aligned to the next PO-MSA, aligned identical residues are fused whether

they are nodes or edges and aligned non-identical residues are recorded as aligned. Finally, any edges connecting the same pair of nodes are removed.

2.6.3 T-COFFEE

T-COFFEE (Tree-based Consistency Objective Function For alignmEnt Evaluation) (Notredame et al., 2000) is a global progressive consistency-based algorithm. Initially, all pair-wise alignments of the sequences are performed twice: once with the global alignment method ClustalW (Thompson et al., 1994) where a single global alignment is generated and once with the local alignment method Lalign (Huang et al., 1990) where 10 top-scoring non-intersecting local alignments are generated. The results are pooled into a primary library of combined weights for each non-redundant residue pair. The combined weight for each residue pair (x, y) corresponds to the sum (Σ) of scores (S) of the global and local alignments containing that residue pair. Each alignment score (S) is the percentage sequence identity of that alignment. A library extension step is then performed using a procedure called *matrix extension* (Notredame et al., 2000) to measure how residue pairs align with respect to other residues in the library, producing triplet weights. These triplets are then used to assess how well sequences are aligned compared to the other sequences in the dataset, rather than looking at pairs of sequences in isolation. The final alignment is built by performing the library extension step to produce a guide dendrogram, which then orders how the sequences are aligned.

2.6.4 MUSCLE

MUSCLE (MUltiple Sequence Comparison by Log-Expectation) (Edgar, 2004) is a recent global progressive alignment method. The sequences are ordered by using a quick and dirty similarity measure that is clustered into the alignment guide tree by UPGMA. This substantially reduces the running time of the algorithm because the all-against-all pair-wise step is skipped. The tree branches are from that point on iteratively aligned and in the end an optimised alignment of the sequences is produced. One of the innovations of the MUSCLE method is the use of the ‘log-expectation’ scoring scheme for the dynamic programming strategy, briefly described earlier (see section “The MSA scoring function”). The iterative optimisation step and the reduction in computational

complexity of the initial steps of the alignment make MUSCLE one of the fastest and most accurate progressive alignment methods currently available.

2.7. ASSESSMENT OF MSA

In this section we discuss MSA benchmarking issues and the scoring schemes currently in use to evaluate MSA quality using reference alignments. As high alignment scores do not necessarily entail a good biological quality, we also briefly discuss MSA score optimisation.

2.7.1 Evaluating multiple sequence alignments methods

Evaluating MSA programs is a complex issue. First of all, there is no general agreement as to what the standard of truth should be. For instance, should an alignment be evaluated using evolutionary, structural, or functional criteria? Although in closely related familial sequences these criteria are expected to lead to the same alignment, in more distant cases they can result in very different answers. Moreover, benchmarks are usually carried out using a set of reference alignments, so that the evaluation becomes crucially dependent on the quality of such a reference alignment database. A few recent attempts to alleviate this database problem are based on using protein 3D structures directly in assessing the alignments (O'Sullivan et al., 2003). Furthermore, different ways have been proposed to quantify the agreement between a proposed and a reference alignment, such as un-weighted or weighted SP scores, or the column score. The un-weighted SP score implies checking for each aligned amino acid pair in the reference MSA whether this pair has also been aligned in the alignment produced by the method considered. The final score usually is the percentage of the total number of aligned pairs in the reference alignment that have also been matched in the query alignment. The weighted SP score follows basically the same protocol but weights each pair with the corresponding value from an amino acid exchange matrix (e.g. BLOSUM62). Finally, the column score checks for each column in the reference alignment whether the amino acids found aligned here have been reproduced exactly in the query alignment: if only one sequence is misaligned at the column considered, the whole column is taken to be incorrectly reproduced. Compared to the un-weighted and weighted SP scores, the column score is a more stringent measure for alignment

evaluation. For example, an outlier sequence that is distant from all other sequences in the query set has a relatively high chance of becoming misaligned, and this will be reflected much more dramatically in the column score than in either SP scores.

In the sections below we describe how to evaluate a MSA in the absence of a reference and what scoring schemes can be used when a reference is present. We will also briefly describe the most frequently used MSA benchmarking datasets and other alignment sets that can also be used.

a. Evaluation without a reference MSA

Since pair-wise alignment algorithms optimise residue exchange scores and gap penalties, an obvious way of scoring multiple alignments is to extend the pair-wise sequence scores to get a single score for a multiple alignment. This is referred to as the Sum-of-Pairs (SP) score for alignment: for each amino acid $a_{i,j}$ in sequence i and at position j in the multiple alignment, the SP score is $S(j) = \Sigma_{k<l}\, s(a_{k,j}, a_{l,j})$, where $s(a_{k,j}, a_{l,j})$ is the amino acid exchange value. Using the SP alignment column scores, alignments are scored by taking the total sum of the SP scores: $S = \Sigma_{1\leq j\leq N} S(j)$, where N is the number of aligned positions. In the early simultaneous multiple alignment method MSA (Lipman et al., 1989), each cell in the multi-dimensional search matrix, which is corresponding to a column in a resulting multiple alignment, is scored with the SP score. This requires special gap handling for those matrix cells associated with gaps. Here, the SP score of a multiple alignment is calculated without extra treatment of gaps, consistent with the fact that gaps are also ignored in the evaluation against benchmark MSAs, described next.

b. Evaluation against a reference MSA

There are two main schemes for comparing a proposed to a reference MSA: the column score and the sum-of-pairs score (different to the SP score mentioned in the previous section).

The column score of a MSA is calculated by comparing the alignment columns of the proposed MSA with those in the corresponding reference MSA and only taking as correct the ones that are identical. This is a more salient measure than the sum-of-pairs (SP) scores, where over all observed aligned amino acid pairs in a reference MSA,

the fraction of those observed in the corresponding target MSA is compiled. Whereas a single misaligned sequence can zero the column score, the SP score only gradually goes down with more misaligned sequences. Note that the SP scoring system here involves two MSAs, and is therefore different than the aforementioned SP scoring system for a single MSA without a reference.

2.7.2 MSA Standards of truth

The most frequently used reference MSA database is BAliBASE (Benchmark Alignment dataBASE) (Thompson et al., 1999a; Bahr et al., 2001). What has made BAliBASE so appealing to MSA method developers is that it is the only reference MSA database that has been specifically designed for MSA benchmarking. As a result it has been used in many studies as the standard of truth for comparing the performances of new MSA methods with older ones (Thompson et al., 1999b; Notredame et al., 2000; Karplus and Hu, 2001; Heringa, 2002; Edgar, 2004). The BAliBASE alignments are manually verified and corrected by super-positioning of all known 3D structures (Bahr et al., 2001). The current version of BAliBASE (version 2.0) contains a total of 167 reference alignments (December 2003) placed in eight different categories, which are aimed at covering most of the problems alignment engines come up against: (1) MSAs containing equidistant sequences of various conservation levels, (2) alignments with a single orphan sequence, (3) alignments comprising two distant groups of less than 25% sequence identity, (4) alignments containing long insertions, (5) alignments containing long deletions, (6) sequence repeats, (7) transmembrane sequences and (8) domain permutations.

However, there are other structural alignment databases that can serve as reference sets although their development has not been intended for MSA benchmarking. HOMSTRAD (HOMologous STRucture Alignment Database) is a database of aligned three-dimensional structures (Mizuguchi et al., 1998; de Bakker et al., 2001). It contains both pair-wise as well as multiple alignments of sequence families that have been grouped together based on their sequence and structural similarity. The organisation of homologous families is achieved by manual editing and complemented by automated structure comparison methods. The uniqueness of

HOMSTRAD is that it provides annotation of the conserved structural features of the sequences in each alignment.

The SCOP (Structural Classification Of Proteins) database (Murzin et al., 1995; Hubbard et al., 1998) has released a structural alignment database of its families, called PALI (Sujatha et al., 2001). The PALI database provides both pair-wise and multiple structure-based sequence alignments for homologous proteins of known 3D structure. The database also provides dendrograms depicting phylogenetic relationships based on sequence and structural similarities. More recently, further initiatives have been developed for the assessment of multiple alignment methods (Schultz et al., 2000; Edgar, 2004; Van Walle et al., 2004).

2.7.3 Optimising alignment scores

The SP scores of 'incorrect' sequence alignments are often higher than those of 'true' reference alignments derived from structural super-positioning of proteins. This is caused by a lack of structural information in the substitution models. Heringa (2002) calculated SP scores (for single alignments) for each of the BAliBASE benchmark alignments (Thompson et al., 1999a). These scores were then compared to corresponding SP scores of the alignments calculated using non-optimised Praline conditions. More than a quarter of the Praline alignments turned out to have higher SP scores than the corresponding reference alignments, while for the largest BAliBASE alignments, more than half attain larger SP scores for the default PRALINE alignments than those of the corresponding reference alignments. This might be referred to as the "Charlie Chaplin" problem: At the peak of his fame, Charlie Chaplin allegedly entered a Charlie Chaplin contest *in cognito* and was ranked second (Heringa, 2002). It is clear that trying to optimise the SP score for alignments that already score higher than their corresponding reference alignments is not likely to lead to convergence to the latter alignments.

New avenues to novel scoring schemes as well as benchmarking methods are being actively researched. Lin *et al*. (2003) introduced a new alignment-scoring scheme CAO (Contact Accepted mutaiOn) based on the amino acid interactions in tertiary structures. The scheme is based on a new evolutionary model expressed in 400×400 residue contact mutation matrices, and can be used to evaluate alignments whenever

there is a tertiary structure at hand for one or more of the sequences, from which the pair-wise residue contacts can be derived. Since the contact-based evolutionary model combines sequence and structure information, it yields biologically more meaningful alignment scores (Lin et al., 2003).

Chapter 3

Secondary Structure Prediction and Its Co-Dependence with Multiple Sequence Alignment

The content of this chapter contains work published in Simossis VA, Heringa J (2004) Integrating protein secondary structure prediction and multiple sequence alignment. Curr Protein Pept Sci 5:249-266 and Simossis VA, Heringa J (2005) Local structure prediction of proteins. In: Xu Y, Xu D, and Liang J (eds). Computational methods for protein structure prediction and modeling. Springer Verlag, Chapter 8.

3.1. SECONDARY STRUCTURE BASICS

A secondary structure element is a section of consecutive residues in a protein sequence that corresponds to a local region in the associated protein tertiary structure and shows distinct geometrical features. The two basic secondary structure types, *the α-helix* and *β-strand*, are regular and easily distinguishable in protein tertiary structures (Figure 3.1), while other types are sometimes harder to classify. For this reason, the majority of secondary structure prediction methods use a three-class alphabet for their predictions: α-helix (H), β-strand (E) and other; the latter often referred to as *coil* (C).

Approximately 50% of the amino acids in all known proteins are associated with either α-helices or β-strands, while on average the remaining half of protein secondary structure is irregular. The primary reason for the regularity observed in helices and strands is the innate polar nature of the protein backbone, which comprises a polar nitrogen and oxygen atom in each peptide bond between two successive amino acid residues. For a protein to become foldable with an acceptable internal energy, the parts of the backbone buried in the internal protein core need to form hydrogen bonds between these polar atoms. The α-helix and β-strand conformations are optimal for this, since each nitrogen atom can associate with an oxygen partner (and *vice versa)* within and between both secondary structure types. However, in order to satisfy the hydrogen-bonding constraints, β-strands need to interact with other β-strands, which they can do in a parallel or anti-parallel fashion to form a *β-pleated sheet*. As a result, β-strands depend on crucial interactions between residues that are remotely situated in the sequence and therefore are believed to have more pronounced context dependencies than α-helices. Consequently, most prediction methods have greatest difficulty in predicting β-strands correctly.

3.2. BIOCHEMICAL FEATURES OF SECONDARY STRUCTURES USED IN PREDICTION

Analyses of secondary structure and related features of the many protein structures deposited in the Protein Data Bank (PDB) (Berman et al., 2000) have resulted in a set of rules about α-helices, β-strands and coil structures that are important

for secondary structure prediction. Most prediction methods, either implicitly or explicitly, make use of these observations when performing their predictions.

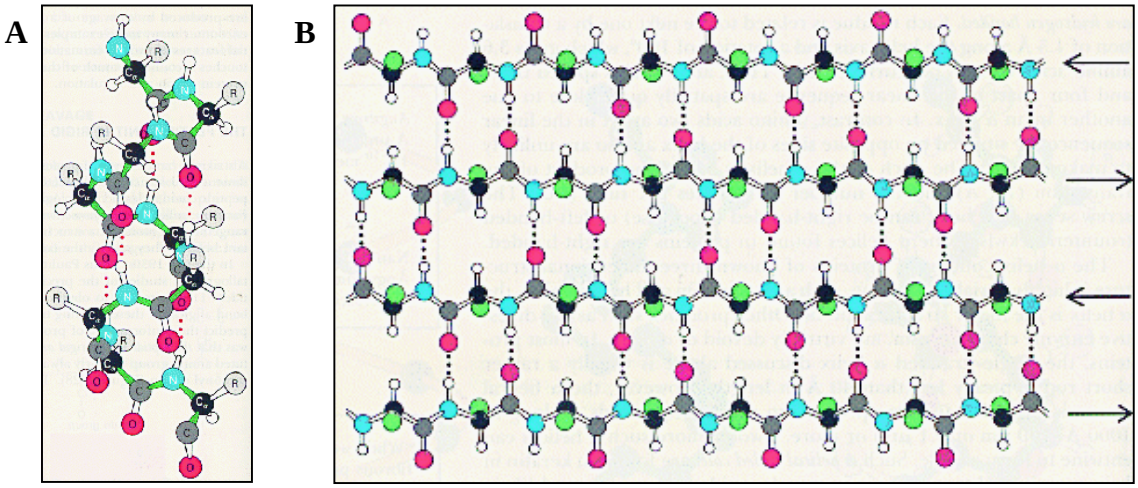


Figure 3.1. Images of A) the alpha-helix and B) the beta pleated sheet structures showing H-bonds.

3.2.1 α-helices

Considering that ideally one turn of the helical structure is made up of 3.6 residues, the minimum predicted length for a α-helix should be three or four residues. Also, α-Helices are often positioned against a buried protein core and have one phase contacting core hydrophobic amino acids, while the opposite phase interacts with the solvent. This results in so-called amphipathic helices (Schiffer and Edmundson, 1967), which show an alternating pattern of three to four hydrophobic residues followed by three to four hydrophilic residues. As an additional rule, proline residues are rare in middle segments as they disrupt the α-helical turn, while they are more frequent in the first two positions of the structure.

3.2.2 β-strands

Normally, two or more β-strands constitute a β-pleated sheet with two strands forming either edge. The hydrophobic nature of such edge strands is different from that of strands that are positioned inside the sheet because they are shielded on both sides. As side-chains of constituent residues along a β-strand alternate the direction in which they protrude, edge strands of a β-sheet can show an alternating pattern of

hydrophobic-hydrophilic residues, while buried strands typically comprise hydrophobic residues only. The β-strand is the most extended conformation (i.e. consecutive Cα atoms are farthest apart), so that it takes relatively few residues to cross the protein core with a strand. Therefore, the number of residues in a β-strand is usually limited and can be anything from two or three amino acids. Further, β-strands can be disrupted by single residues that induce a kink in the extended structure of the backbone. Such so-called β-bulges consist of relatively hydrophobic residues.

3.2.3 Coil structures

Multiple alignments of protein sequences often display gapped and/or highly variable regions, which would be expected to correspond to loop (coil) regions rather than the other two basic secondary structures. Loop regions contain a high proportion of small polar residues like alanine, glycine, serine and threonine. Glycine and proline residues are also seen in loop regions, the former due to their inherent flexibility, and the latter for entropic reasons relating to the observed rigidity in their kinking the backbone.

3.3. SECONDARY STRUCTURE PREDICTION: THE BEGINNING

The use of computers to predict protein secondary structure started just over thirty years ago (Nagano, 1973). All computational methods devised early on based there predictions on single protein sequences and the average prediction accuracy lingered for a long time in the range between 50-60% correctness, i.e. 50-60% of the residues used for predictions were correctly assigned a secondary structure class H, E or C (Schulz, 1988). A random prediction would yield about 40% correctness given the observed distribution of the three states in globular proteins, i.e. 30% α-helix, 20% β-strand and 50% coil. Although significantly beyond the random level, the accuracy of the early prediction methods was not sufficient to allow the successful prediction of protein topology, i.e. the folded structural arrangement of protein secondary structures.

The pioneering algorithms of Nagano (Nagano, 1973) and Chou and Fasman (Chou and Fasman, 1974) were aimed at predicting the secondary structure for single sequences and relied on a statistical treatment of compositional information. Lim's

method (Lim, 1974a) represented the first attempt to incorporate stereochemical rules in prediction. The method relied mainly on conserved hydrophobic patterns in secondary structures such as amphipathicity in helices (Schiffer and Edmundson, 1967). The early and popular GOR method (Garnier et al., 1978; Gibrat et al., 1987) considered the influence and statistics of flanking residues on the conformational state of a selected amino acid to be predicted. The popular early methods by Nagano (Nagano, 1973), Lim (Lim, 1974b), Chou-Fasman (Chou and Fasman, 1974) and the GOR method (Garnier et al., 1978; Gibrat et al., 1987) were reported to perform single sequence secondary structure prediction with accuracies of 50%, 54%, 56% and 64.4% (GOR IV; (Garnier et al., 1996)), respectively.

3.4. FROM EARLY TO RECENT PREDICTION: THE KEY ADVANCES

The first important breakthrough for secondary structure prediction was the use of multiple sequence alignment (MSA) information (Dickerson et al., 1976), which was incorporated into an automatic prediction method for the first time by Zvelebil *et al.* (Zvelebil et al., 1987). The use of the evolutionary information stored in a MSA of a family of homologous proteins, as opposed to using a single sequence, is essential for more accurate predictions and as a result, all current state-of-the-art secondary structure prediction methods use MSAs.

Secondly, the use of increasingly sensitive machine-learning techniques made the translation process of the evolutionary information in MSAs more accurate. Since the 1990s, methods have employed various complex decision-making techniques including neural networks (NNs); k-Nearest-Neighbour analysis (kNN); Example Based Learning (EBL); Hidden Markov Models (HMMs); and Support Vector Machines (SVMs). An overview of these techniques is provided later on (see section “State-of-the-art secondary structure prediction techniques”).

The third element that allowed secondary structure prediction methods to rapidly advance was the dramatic increase in protein sequence and structure data, combined with the enhanced sensitivity of automatic database searching tools (Altschul et al., 1997; Friedberg et al., 2000). This allowed the correct identification of more divergent homologues and subsequently the creation of larger structural family profiles that encapsulate more divergent information. In addition, this increase in information

also allowed the training of machine-learning algorithms on larger data sets, resulting in higher method accuracy and sensitivity.

As a result, the three standard steps used by almost all current secondary structure prediction methods are: (i) detecting homologues from a database for the sequence to be used as input, (ii) aligning these sequences, and (iii) using the position-specific information in the MSA to predict the secondary structure of the input

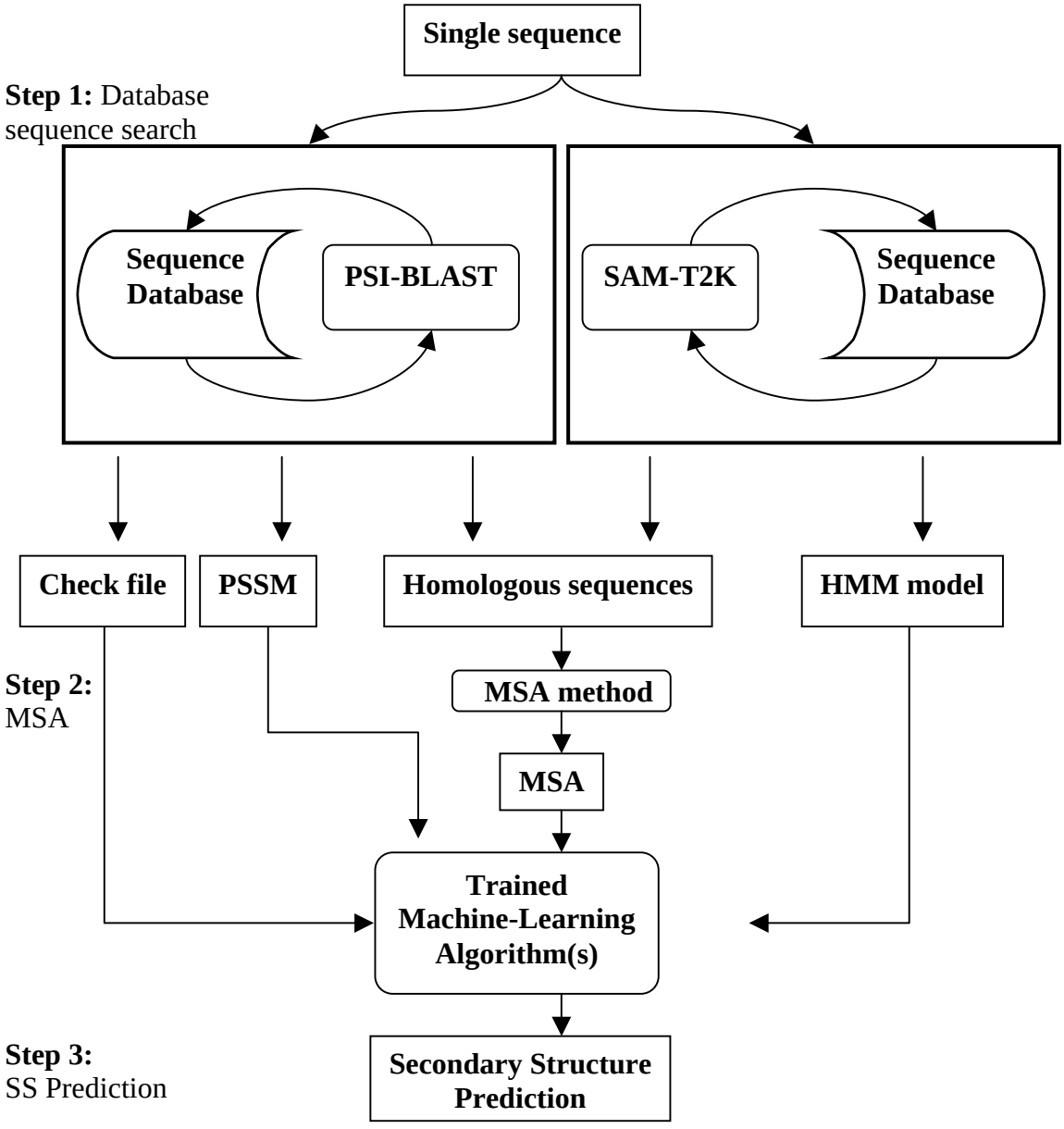


Figure 3.2. The currently employed three-step process that leads to secondary structure prediction. Step 1: sequence database searching (here we show the currently top methods PSI-BLAST and SAM-T2k); Step 2: multiple sequence alignment (MSA) of the selected sequences either in the possible output formats of the database search methods or by separately employed MSA methods; Step 3: secondary structure prediction based on one of the MSA types of Step 2.

sequence (Figure 3.2).

3.5. DATABASE SEARCHING AND SECONDARY STRUCTURE PREDICTION

Obtaining the sequences to compile an MSA is done in two main ways: either the MSA is made from already selected homologous sequences or a database homology search engine is used with the query sequence as input to identify homologous sequences in sequence databases. In the latter case, an MSA method is then used to align the query and homologous sequences.

Since the successful first use of the PSI-BLAST database search tool (Altschul et al., 1997; Altschul and Koonin, 1998) in the prediction method PSIPRED (Jones, 1999), most newly developed, but also older prediction methods (such as PHD (Rost and Sander, 1993) that was updated to PHDpsi (Przybylski and Rost, 2002)), have followed in the same footsteps and use PSI-BLAST to produce their input MSAs.

3.6. STATE-OF-THE-ART SECONDARY STRUCTURE PREDICTION TECHNIQUES

The most popular machine learning approaches used in current secondary structure prediction methods include *k*-nearest-neighbour analysis (kNN), artificial neural networks (NNs), hidden Markov models (HMMs) and support vector machines (SVMs). Each technique offers unique advantages and also has associated drawbacks in tackling complex problems such as pattern recognition, which for our purpose is the identification of structural classes from consecutive residue patterns. In the descriptions to follow, we give a basic overview of each technique and discuss their strengths and weaknesses.

3.6.1 *K*-nearest-neighbour

The *k*-Nearest-Neighbour (kNN) technique is an instance-based machine learning technique. In order to predict the secondary structure of a protein, for which the structure is unknown, the technique extrapolates from already existing information of related proteins. As a result, the performance of this approach is directly dependent on whether closely related examples of known secondary structure are available.

Assuming enough "related" information is available, kNN has distinct advantages over other methods: it is easy to program and can deal with complex problems using low complexity approximations; it can deal with noisy data; it involves no training or re-training with new data and never loses information content because all learning material is explicitly used every time the method is run.

The prediction main steps involve the creation of a library of protein "fragments" of known secondary structure, the creation of a distance representation scheme for relating the library fragments to the query and a decision scheme for discerning between multiple matching possibilities. The various k-nearest-neighbour prediction methods approach these steps differently. As a simple example, let *X* be a protein sequence for which we want to do a prediction (Figure 3.3). Let our example method have access to a large, non-redundant database of variable-length protein sequence fragments with their corresponding secondary structures. Our method would take sequence *X* and use it to scan the database for related fragments (neighbours). Now, let our measure of "relatedness" be the local alignment score between our sequence *X* and each fragment. The higher the score, the more related the fragment. After identifying potential neighbouring fragments, we would sort them and use only the *k* nearest ones for the prediction to minimise errors and processing time, as long as we ensured that *k* allowed good coverage across the whole length of *X*. Finally, for all sequence positions where more than one possible secondary structure was present, our decision scheme could be a "majority vote" consensus, where the most prominent secondary structure is assigned. Here, each possibility could be further weighed in relation to the fragment's distance from *X*, making the closest fragments count more than less related ones. The string of these decisions would be our prediction. Again, at this point we could apply a filter to "tidy up" the prediction, for example correcting impossible structures such as a single residue helix. In any case, the possibilities are many and this was merely intended as a guide example for how a kNN prediction works.

kNN methods out-perform classical neural network (NN) methods when closely related examples to a query are available, but their success is highly restricted as they perform poorly in all other cases. The huge increase in data availability has provided the kNN approaches with larger, more diverse sets of examples to train on, thus

increasing the space in which they accurately perform. Nonetheless, the data sets that are currently available are still far from covering the entire protein universe. As a result, the NN approach is still the best way to predict secondary structure in a wide range of cases. As a solution to this, methods have been developed over the years that attempt to combine the best of both worlds, an example of which is APSSP2 (Raghava, 2002).

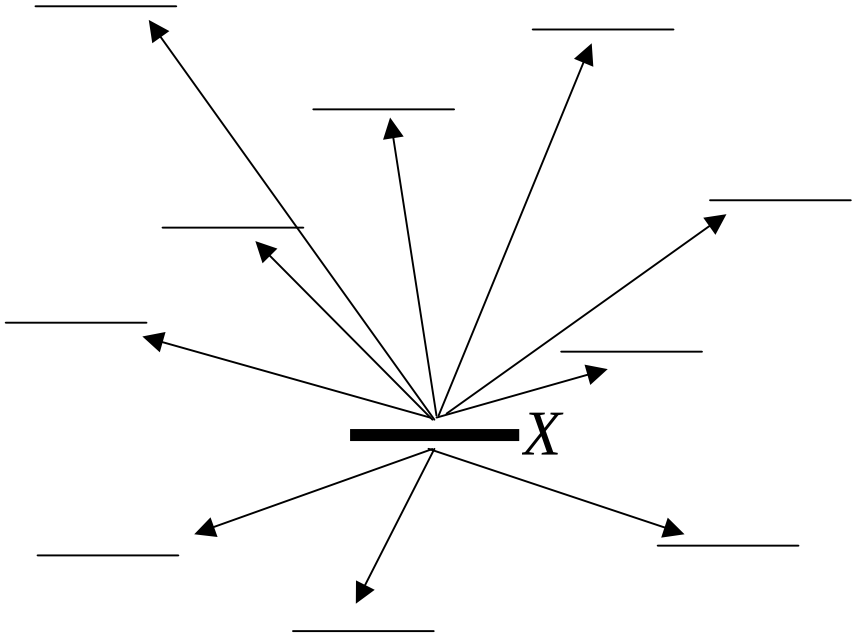


Figure 3.3. The kNN approach to classifying the secondary structure of a sequence fragment based on a database of other fragments with known secondary structures. The fragment *X* under consideration in this case is represented by a thick black line and the surrounding lines represent the database sequences being assessed for relatedness (long arrows correspond to small relatedness).

3.6.2 Neural networks

Neural networks (NN) are complex machine-learning systems that are based on non-linear statistics. They consist of multiple inter-connected layers of input and output units, and can also contain intermediate (or "hidden") unit layers (for a review, see (Minsky and Papert, 1988)). Each unit in a layer receives information from one or more other connected units and determines its output signal based on the weights of the input signals (Figure 3.4). The weights of a NN are chosen depending on the training procedure and the training set. The training procedure is done by adjusting the weights of the internal connections to optimise the grouping of a set of input patterns into a set

of output patterns. In other words, a NN tries to encapsulate the basic trends of the training set (usually a large number of non-redundant examples) and apply them to unknown cases. NNs are powerful learning tools, but there is a risk of over-training the network, which leads to proper recognition of those patterns the NN has been confronted with during training, but much less successful recognition of patterns that have not been seen. For this reason, training sets must be large in number and non-redundant so they can capture a representative sample and thus decrease their bias towards specific cases. More importantly, when testing a trained NN, the test set must be absolutely separate from the training set and as divergent as possible so that the testing is objective and as unbiased as possible. NNs are very common in secondary structure prediction and are used by all top-performing methods, whether in combination with other systems or on their own.

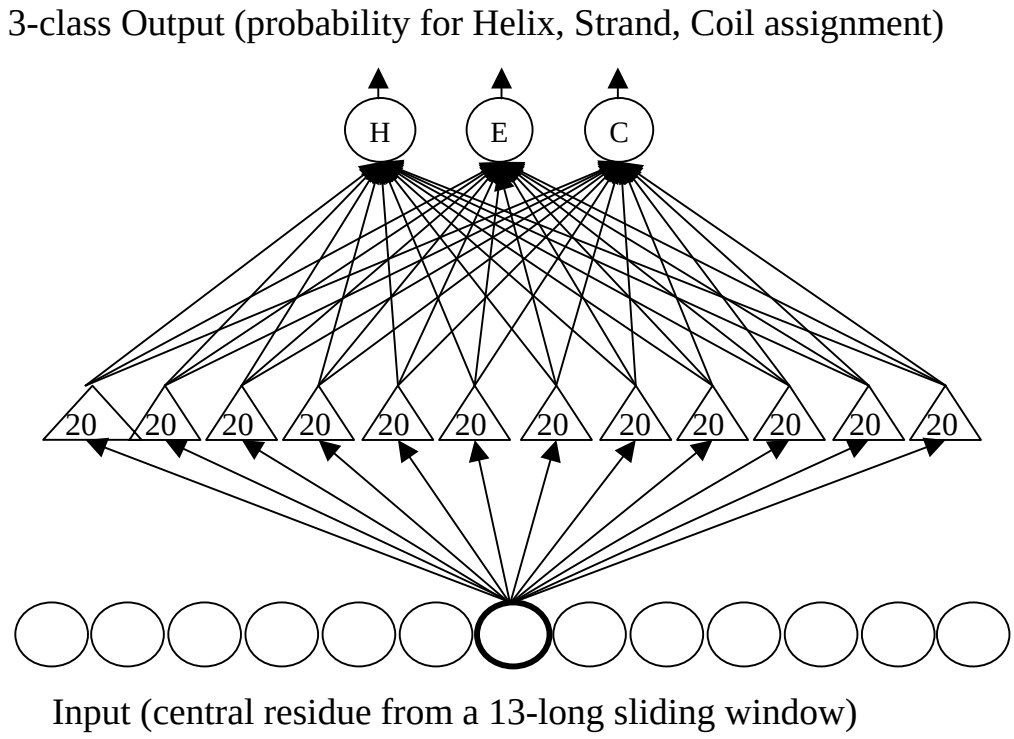


Figure 3.4. A generic schematic representation for secondary structure prediction NN using a window of 13 residues. The number 20 in the hidden layer positions represent the 20 amino acid possibilities for each position of the 13-long window with respect to the central residue. The path through the network outputs a value for the central residue being either helix (H), strand (E) or coil (C).

3.6.3 Hidden Markov models

Hidden Markov Models (HMMs) are a class of probabilistic models usually applied to time series or linear sequences (for reviews see (Eddy, 1996, 1998; Durbin et al., 2000)). They were first introduced to Bioinformatics in the 1980's (Churchill, 1989) and have been applied as protein profile models in the last decade (Krogh et al., 1994). The basic structure of an HMM is a series of *states* that are linked together through *state transitions*. Each of these states also has a symbol emission probability distribution for generating a symbol in the alphabet. For example, let us consider a sequence modelling HMM that describes 3 possible amino acid states, according to what secondary structure element (SSE) they are in (Figure 3.5). In any point after the HMM is initialised, we are in a state X (helix, strand or coil) and have the possibility of either switching to a different state Y or remaining in the same state X. The decision for this is governed by the relation between state transition probability from state X to state Y, and from state X to X. In addition, when the transition is made the HMM will emit (generate) a character from the alphabet (in this case the SSE symbols H, E or C) with the probability linked to that state. This process is repeated until an end state is reached. In the end there are two layers in our HMM, a hidden *state sequence* that we do not see and a *symbol sequence* that we do see.

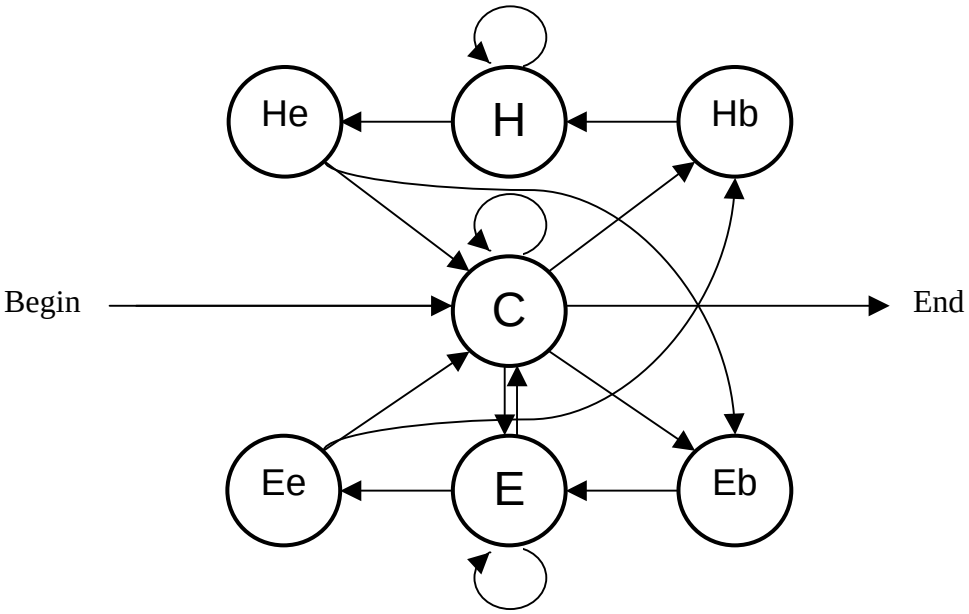


Figure 3.5. A HMM for secondary structure prediction (Lin et al., 2005) (Chapter 4). He and Ee are helix and strand end positions, respectively; Hb and Eb are helix and strand beginning positions, respectively; and H and E are all other helix and strand positions.

A HMM can be parameterised by either training or building procedures. In the training procedure of a sequence-structure prediction HMM, a set of unaligned sequences would be used, while in the building procedure, a set of pre-aligned sequences would used. It is generally advisable to build HMMs whenever there is reference information possible.

The HMMs that are used in sequence database searching and structure prediction are profile HMMs. A profile HMM is a strictly linear, left-to-right model that comprises a series of nodes, each corresponding to a column in a multiple alignment (Krogh et al., 1994). Each node has a match state, insert state and delete state. Each sequence uses a series of these states to traverse the model from start to end. Using a match state indicates that the sequence has a character in that column, while using a delete state indicates that the sequence does not. Insert states allow sequences to have additional characters between columns. In many ways, these models correspond to profiles. The primary advantage of these models over standard methods of sequence search is their ability to characterize an entire family of sequences.

3.6.4 Support Vector machines

The Support Vector machine (SVM), first introduced by Vladimir Vapnik in 1992, is a linear learning machine based on recent advances in statistical theory (Vapnik, 1995, 1998). In other words, the main function of SVMs is to classify input patterns by first being trained on labelled data sets (supervised learning). SVMs have been shown to be a significant enhancement in function compared to other commonly used machine learning algorithms such as the perceptron algorithm (see section “Neural Networks”) and have been applied to many areas such as handwriting, face, voice and object recognition and text characterization (for a comprehensive description of SVMs see (Cristianini and Shawe-Taylor, 2000)). With the turn of the millennium, SVMs were extensively applied to classification and pattern recognition problems in bioinformatics (for reviews see (Byvatov and Schneider, 2003; Noble, 2004)).

The power of SVMs lies in their use of non-linear kernel (similarity) functions. When a linear algorithm such as the SVM uses a dot product, replacing it with a non-linear kernel function allows it to operate in different space. Hence, the kernel functions used in SVMs implicitly map the input (training or test data) into high-dimensional

feature spaces. In the high-dimensional feature spaces, linear classifications of the data are possible (each classifier is a separate dimension); they become non-linear in following steps where they are transformed back to the original input space. As a result, although SVMs are linear learning machines regarding the high-dimensional feature spaces, in fact they act as non-linear classifiers.

The key is to carefully design the kernel (similarity) criteria during training so that it will best discriminate each class (for more information on kernels used in computational biology see (Schoelkopf et al., 2004)).Ultimately, the kernel function generates a maximum-margin hyperplane between two classes and resides somewhere in space. For example, if we were training an SVM for helix prediction, given training examples labelled either "helix" or "non-helix", our kernel function would generate a maximum-margin hyperplane that would split the "helix" and "non-helix" training examples so that the distance from the closest examples (the margin) to the hyperplane would be maximized. If the hyperplane is not able to fully separate the "helix" and "non-helix" examples, the SVM will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. The parameters of the maximum-margin hyperplane are derived by solving a quadratic programming (QP) optimisation problem. The examples closest to the hyperplane (decision boundary) are “support vectors”, while the ones far from it have no effect. After training, any “unknown” input for which we want to decide whether it is helix or not is mapped into the high-dimensional space and the SVM decides whether it is “helix” or “non-helix”. However, since secondary structure elements are usually classified in three states (helix (H), strand (E) and coil (C)), the actual recognition challenge is not binary (helix or non-helix), but multi-class and therefore the prediction is still incomplete. The multi-class recognition problem is tackled differently across SVM prediction methods (Hu et al., 1997; Hua and Sun, 2001; Kim and Park, 2003; Ward et al., 2003; Guo et al., 2004).

3.6.5 Consensus secondary structure prediction

The majority of secondary structure prediction methods are trained using information from proteins of known 3D structure. In modern studies, training is performed on large datasets, thus avoiding over-fitting, and the training data sets do not

include any of the proteins used to assess the final version of the method (jack-knife testing). However, each method is trained on different sets of proteins and as a consequence this introduces a bias to the prediction performance, depending on the type of proteins used in the training set.

An early attempt to minimise these biasing effects was to combine predictions from various methods to produce a single consensus (Cuff et al., 1998; Cuff and Barton, 1999). The consensus was derived by majority voting, where the per-residue predicted states from each method were given each an equal "vote" and the consensus kept the prediction that got the majority of the "votes". The philosophy of deriving a consensus prediction is similar to the that of having 3 clocks on a boat: if one clock shows the wrong time there are always the other two to check for consistency and since the probability that two out of three clocks will go wrong at the same time and in a similar way is very low, it is a safe assumption to go with the majority. During the same time, other strategies for consensus prediction were developed such as the combination of different neural network outputs (Chandonia and Karplus, 1999; Cuff and Barton, 2000; King et al., 2000; Petersen et al., 2000); optimal method choice for the consensus scheme by linear regression statistics (Guermeur et al., 1999) and decision trees (Selbig et al., 1999); deriving a consensus from cascaded multiple secondary structure classifiers (Ouali and King, 2000); and expressing the consensus as a composite predicted secondary structure, where the variation in prediction is not resolved but used as extended information for the successive database searching steps for fold recognition (An and Friesner, 2002).

From these consensus-deriving strategies, the "majority voting" consensus-deriving scheme has been used in recent investigations using more state-of-the-art predictions methods and the results have consistently shown that a consensus prediction is better than any of the single predictions produced by the methods used for deriving the consensus (Albrecht et al., 2003; McGuffin and Jones, 2003; Ward et al., 2003).

3.6.6 Tertiary structure feedback for secondary structure prediction

In the prediction techniques we have described up to now, predicting the secondary structure of a protein from its amino acid sequence has mainly involved using adjacent information. However, when a protein folds, the secondary structure

elements that were initially formed can be influenced by the dynamics of formerly distant regions, which now have been brought closer due to the structural re-arrangement in three-dimensional space (Blanco et al., 1994; Ramirez-Alvarado et al., 1997; Reymond et al., 1997). Although many initial conformations remain unchanged in the folded protein, there are regions that undergo transitions from one secondary structure element type to another as a result of different types of interactions (Minor and Kim, 1996; Cregut et al., 1999; Luisi et al., 1999; Derreumaux, 2001; Macdonald and Johnson, 2001). As a result, even the best prediction methods make wrong predictions for these cases because the transition changes only happen as a result of tertiary structure interactions and have not yet occurred in the un-folded state.

Meiler and Baker (2003), used low resolution tertiary structure models to feed back three-dimensional information to the predictions and successfully raised the quality of the predictions, particularly in β-strands (Meiler and Baker, 2003). However, the applicability of the method is limited since it is only applicable to single-domain proteins and is not able to account for inter-domain interactions.

In another approach, surface turns that change the overall direction of the chain ("U" turns) were predicted using multiple alignments and predicted secondary structure propensities to improve the quality of the predictions (Hu et al., 1997; Kolinski et al., 1997).

3.7. EVALUATING SECONDARY STRUCTURE PREDICTION METHODS

Let us assume that we have developed a new secondary structure prediction algorithm and we want to test it. How do we evaluate its performance so that it can be comparable to con-current methods? The evaluation of secondary structure prediction needs at present three main components: a standard of truth or reference structure, a useful scoring scheme, and a standard of evaluation. The standard of truth is used to find out how good is the prediction a method produces. So the standards of truth are databases of sequences with known secondary structure, usually derived from the spatial co-ordinates of 3D structures solved by NMR or X-ray crystallography. The second component is a way to informatively compare the prediction with the standard of truth. Finally, an evaluation standard is needed so that a method can be tested without biasing the results in its favour and also so that the results can be comparable

between other rival methods. This is achieved by using test sets that are not included in the methods training and more importantly running all methods on the same sets. In the next sections we will describe the currently used standards of truth; different scoring schemes and their importance; and the currently organised evaluation standards used to assess secondary structure prediction.

3.7.1 Secondary structure standards of truth

At present, the main sources of reference secondary structures are derived from the Protein Data Bank (PDB) (Berman et al., 2000). The PDB is a continually updated database of all available experimentally derived three-dimensional protein structures. The PDB data is in the form of 3-dimensional coordinate files, which can be parsed to extrapolate the secondary structure elements. The most commonly used parser is the DSSP program, which is used to produce the DSSP database (Dictionary for Secondary Structure of Proteins) (Kabsch and Sander, 1983). The DSSPcont (Carter et al., 2003) database is continuously updated with the new PDB entries. Other parsers include STRIDE (Frishman and Argos, 1995) and DEFINE (Richards and Kundrot, 1988).

3.7.2 Secondary structure prediction evaluation standards

There are two ways one can address the issue of protein structure prediction accuracy. The first way is from the developer's point of view, where the interest is in how well a method can react to a challenging problem. This question is addressed in the CASP and CAFASP meetings. On the other hand, the second way is from the user's point of view, so mainly molecular biologists. Here the interest is in which method is overall better, so that misleading results can be minimised. In this case, the EVA team has set up a server that continually evaluates the accuracy of prediction programs that are registered to it.

The CASP experiments (Critical Assessment of techniques for protein Structure Prediction) are organised to assess all types of methods for predicting protein structure and discuss the current advances in the field and future directions and improvements for problematic areas of the field. The first CASP meeting was held in 1994 and has since been held bi-annually in different locations around the globe. The most recent experiment was CASP 6, at the end of 2004. The CASP experiments put together

protein sets of which the solved structural information has not been released yet and challenge all methods that take part to do their best predictions. This way, every two years the best methods compete on the same grounds on newly solved proteins. In addition, closely linked to CASP are the CAFASP experiments (Critical Assessment of Fully Automated Structure Prediction), which use the CASP protein sets to test automatic prediction servers, which are available online for researchers to use. The fourth and latest CAFASP experiment was held together with CASP 6 in 2004. These experiments are mainly aimed to give an assessment of what online automatic tools are currently available to researchers and to determine how good they are by assessing them on equal terms.

The EVA server (Koh et al., 2003) (EValuation of Automatic protein structure prediction) is a web-based assessment tool at Columbia University (URL) that has been performing evaluations of the accuracies of its member structure prediction servers since June 2000 (Koh et al., 2003). The assessment comprises four different categories of structure prediction: a) comparative modelling, b) fold recognition and threading, c) secondary structure prediction and d) inter-residue contact prediction. Here we will focus mainly on the assessment of secondary structure prediction. The EVA server updates its reference secondary structure datasets on a daily basis by retrieving the most up-to-date experimentally determined structures from the PDB and employing the DSSP program to parse the 3D coordinates into secondary structure chains. The amino acid sequences of the newly acquired proteins are then submitted to the member secondary structure prediction servers and their predictions are evaluated with reference to those generated by the DSSP program. At present (January 2005), the active secondary structure prediction server-members assessed by EVA are APSSP2 (Raghava, 2002), Jpred TNG (Cuff and Barton, 2000), PHD (Rost and Sander, 1993), PHDpsi (Przybylski and Rost, 2002), PROF (Ouali and King, 2000), PROFsec (Rost, personal communication), Prospect (Kim et al., 2003; Kim and Park, 2003), PSIPRED (Jones, 1999), SAM-T99sec (Karplus et al., 1999; Karplus and Hu, 2001), SSPRO (versions 2, 4 and SCRATCH) (Pollastri et al., 2002), SABLE (versions 1 and 2) (Adamczak et al., 2004), and JUFO (Meiler et al., 2001). The secondary structure prediction method YASPIN (Lin et al., 2005) that was developed during this project will be described in Chapter 5 and is one of the new added members. Table 3.1 contains

a list of the current online URLs these programs can be used at and the average percentage accuracy they have scored on the respective EVA independent test cases. All assessment results since 1999 are made freely available on the EVA website at Columbia University (http://maple.bioc.columbia.edu/eva/) and are also mirrored at the UCSF (http://eva.compbio.ucsf.edu/~eva/) and at the CNB Madrid (http://pdg.cnb.uam.es/eva/).

3.7.3 Secondary structure prediction evaluation measures

The evaluation of secondary structure prediction is not a trivial issue. Along the years many scoring schemes have been suggested. We describe three types that are most commonly used and have become the standard by which secondary structure is currently evaluated.

Table 3.1. The EVA server member secondary structure prediction methods, the number of independent cases they have been tested on and their average accuracy scores on these test sets (valid up to end 2004). Each server can be accessed and used at the URLs sited in the rightmost column.

Method	**Test Set**	**Score**	**Server URL** (assume 'http://' at the start of each address)
APSSP2	393	75.1	www.imtech.res.in/raghava/apssp2/
Jpred	167	72.8	www.compbio.dundee.ac.uk/~www-jpred/submit.html
JUFO	133	68.9	www.jens-meiler.de/jufo.html
PHD	446	72.2	cubic.bioc.columbia.edu/predictprotein/
PHDpsi	440	73.3	cubic.bioc.columbia.edu/predictprotein/
PROF_king	443	72.7	www.aber.ac.uk/~phiwww/prof/
PROFsec	443	75.3	cubic.bioc.columbia.edu/predictprotein/
Prospect	315	71.7	compbio.ornl.gov/cgi-bin/PROSPECT/
PSIPRED	443	76.2	bioinf.cs.ucl.ac.uk/psipred/psiform.html
SABLE	156	76.0	sable.cchmc.org/
SABLE2	99	76.9	sable.cchmc.org/
SAM-T99sec	396	76.0	www.cse.ucsc.edu/research/compbio/HMM-apps/T99-query.html
SCRATCH	217	75.7	www.igb.uci.edu/tools/scratch/
SSPRO2	257	74.3	www.igb.uci.edu/tools/scratch/
SSPRO4	68	78.7	www.igb.uci.edu/tools/scratch/
YASPIN	80	71.0	ibivu.cs.vu.nl/programs/yaspinwww/

a. The Q3 measure (3-state prediction accuracy)

A Q_n measure is the percentage per-residue accuracy of predicting *n* states correctly, with reference to a corresponding standard of truth. The Q_3 measure takes into consideration a 3-state secondary structure representation corresponding to helix (H), strand (E) and coil (C) per-residue states and is the sum of all correctly predicted states over the observed states. This is then transformed into a percentage (see equation 1).

$$Q_3 = \sum_{(ss=H,E,C)} \frac{predicted_{ss}}{observed_{ss}} \times 100 \quad (1)$$

where the sum of all observed states equals the protein sequence length. The Q_3 measure can also be represented as a per-state measure for greater insight into the quality of prediction for each state. These measures are typically referred to as the Q3H, Q3E and Q3C for helix, strand and coil states, respectively.

b. The SOV measure (Segment Overlap measure)

Unlike the per-residue nature of the Q_3 measure, SOV is a measure that attempts to evaluate secondary structure element (SSE) or segment prediction (Rost et al., 1994; Zemla et al., 1999). The philosophy of this scoring measure is to take into account important parameters of secondary structure that are overlooked by the traditional Q3 measure. First of all, the secondary structure of proteins is segmented, but is treated as a residue specific property by the Q3 measure. For example, the prediction of a 20-residue helix instead of two 8-long helices with 4 coil residues separating them would have a Q3H score of 80% (16/20 correctly predicted), a Q3E of 0% and a Q3C of 0%, thus a final Q3 of 80%. Although arithmetically correct, this does not represent the segmentation accuracy of the prediction and would score very high, even for modern standards. Secondly, the Q3 measure deals with SSE ends with the same weight as their core regions. This is a very harsh assessment, since SSE ends suffer from two major uncertainties: a) even homologous sequences with high sequence similarity exhibit a degree of staggered SSE ends and b) even deriving SSE ends from 3D structures varies between secondary structure-deriving methods like DSSP and STRIDE, which act as standards of truth. In light of this, a less strict evaluation of end-positions in predicted SSEs, in relation to their core regions, would give a more meaningful evaluation since

it would concentrate the penalty load on the important regions.

The SOV measure takes into account the type (H, E, C) and location of the predicted SSEs with respect to those in the standard of truth. In addition, it takes into account the aforementioned natural variation of SSE boundaries between homologous proteins and uncertainty factor of SSE-end assignment. The SOV score of a prediction is calculated by:

$$Sov_{ss} = 100 \times \frac{1}{N_{ss}} \times \sum_{s(i)} \left[\frac{\min ov(s_{obs}, s_{pred}) + \delta(s_{obs}, s_{pred})}{\max ov(s_{obs}, s_{pred})} \times length(s_{obs}) \right] \quad (2)$$

where *ss* is either H, E or C (assuming a 3-state assignment); *N* is the sum of the total length of the predicted and observed SSEs of type *ss*; s_{obs} and s_{pred} are SSEs of type *ss* in the observed and predicted secondary structures, respectively; minov is the actual overlap of the predicted and observed SSE (the number of positions that have the same state in the predicted and observed *s*), $\max_{ov}$ is the total extent for which either the observed or predicted *s* has a residue state *ss*; and δ is the accepted variation which guarantees a ratio of 1.0 when segment ends are very similar.

c. Matthews Correlations

Possibly the most accurate measure of prediction accuracy, the Matthews correlation coefficient (MCC) value takes into consideration the amount of under- and over-predictions as well as the amount of correct predictions for each state predicted.

$$MCC_{ss} = \frac{tp_{ss} \times tn_{ss} - fp_{ss} \times fn_{ss}}{\sqrt{(tp_{ss} + fp_{ss})(tp_{ss} + fn_{ss})(tn_{ss} + fp_{ss})(tn_{ss} + fn_{ss})}} \quad (3)$$

where *ss* is the secondary structure state being assessed, *tp* is the number of true positive *ss* predictions (correctly assigned), *tn* is the number of true negative *ss* predictions (correctly not-assigned), *fp* is the number of false positive *ss* predictions (incorrectly assigned) and *fn* is the number of false negative *ss* predictions (incorrectly not-assigned). The total MCC score of a number of states is their geometric mean. For example, a 3-state (H, E, C) prediction method would have the following overall MCC:

$$MCC_{HEC} = \sqrt[3]{MCC_H \times MCC_E \times MCC_C} \quad (4)$$

Although the above three quality measures all deserve their place, historically

the Q3 has been the most quoted measure, probably due to its simplicity. Yet, the SOV score is gaining grounds, as it allows for some biological flexibility. Unfortunately, the MCC scores are not used as consistently as the others in the literature.

3.8. THE INTERDEPENDENCE OF MSA AND SECONDARY STRUCTURE PREDICTION

We have so far discussed the current progress in the fields of secondary structure prediction and earlier in Chapter 2 that of multiple sequence alignment (MSA). It is clear that the use of MSA in combination with sensitive database searching for greater family profiles, vastly improves the accuracy of secondary structure prediction algorithms. However, secondary structure prediction accuracy is directly dependent on the quality of the MSA that is used, as has been shown in numerous accounts where different MSAs of the same set of proteins can yield very different predictions (Levin et al., 1993).

The same is true to a certain extent with the reverse relationship, where secondary structure is used to guide the alignment of a query set of sequences. The philosophy behind this scheme is to align a query set of sequences using their amino acid composition but also incorporate information about the secondary structure elements (SSEs) they are part of. The aim is to provide MSA algorithms with structural information that will keep the insertion of gaps outside regions (segments) of the protein that comprise SSEs and consequently induce structural regions to be aligned in a more segmented fashion. Furthermore, incorporation of structural information becomes very important when the sequences that are being aligned have very low sequence percentage identities (<30%). In these cases, MSA methods fail to produce good alignments due to the diversity of the residue sequence information (Chothia and Lesk, 1986; Heringa, 2000, 2002). Using secondary structure information aids the alignment of these distant homologous proteins because it is a more evolutionary conserved feature and thus its variation through evolution is much lower than that of amino acid residues (Chothia and Lesk, 1986; Sander and Schneider, 1991). Ideally, alignments should be “guided” by DSSP-derived secondary structures, as they are based on 3D structure information. This, of course, would limit the use of this scheme

to proteins of known structure and in that case structural alignments would be preferred anyway. Therefore, secondary structure is employed for the cases that have no solved structure.

The use of predicted secondary structure to guide MSA has not been exhaustively investigated. Early on, Heringa used PREDATOR (Frishman and Argos, 1996, 1997) predictions to guide the alignments of the multiple alignment method PRALINE (Heringa, 1999) and saw improvements in alignment quality when aligning 13 flavodoxins with cheY, a distant signal transduction protein that has very low sequence similarity but shares the same fold as the flavodoxins (Heringa, 2000). In this case, the secondary structure prediction program used did not depend on the MSA

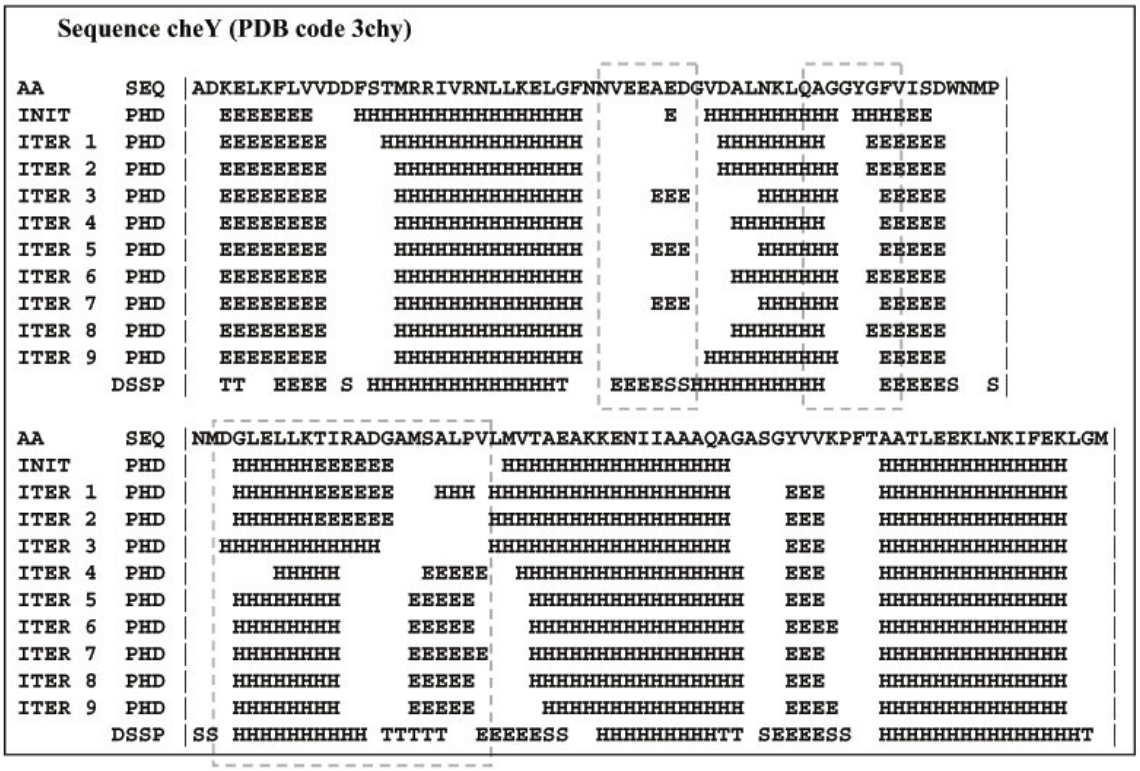


Figure 3.6. The inter-dependence of MSA and secondary structure prediction quality. The AA row represents the top sequence (cheY) of a MSA of 13 flavodoxin sequences. The INIT row is the original prediction produced by PHD on a simple dynamic programming alignment of the sequences. The numbered ITER rows show the influence of iterative secondary structure-guided optimisation of the same alignment using the PRALINE MSA method (see MSA section), on the prediction accuracy of PHD. The DSSP row is the DSSP-derived secondary structure of cheY from the PDB structural information. The regions boxed in grey dotted lines show areas where an increase in MSA quality induces improvements in the accuracy of the predicted secondary structure.

quality. Later on, Heringa (Heringa, 2002) also extended the MSA-secondary structure prediction inter-relationship to an iterative scheme using SSPRED (Mehta et al., 1995), a more advanced MSA-dependant method of the time. In this scenario, an initial MSA is used for the prediction of the secondary structures of the sequences to be aligned and then these predictions are re-introduced to produce a new secondary structure-guided alignment. The new, more correct alignment is then used in the next iteration step to derive new, more accurate secondary structure predictions and so on. During this project we have used PRALINE with the secondary structure prediction methods PHD (Rost and Sander, 1993), PROFsec (Rost, personal communication), JNET (Cuff and Barton, 2000) and SSPRO (Pollastri et al., 2002) in this iterative approach with the earlier tested flavodoxins + cheY orphan sequence alignment case. In Figure 3.6, 10 iteration steps are shown where it is clear that the initial PHD prediction for the most difficult sequence (cheY) is vastly improved by this iterative scheme.

In a more recent investigation, structural information extracted from multiple structure alignment profiles has improved alignment quality between homologues of <30% sequence identity by 17% (Zhang et al., 2003).

Chapter 4

The Influence of Gapped Positions in Multiple Sequence Alignments on Secondary Structure Prediction Methods

The content of this chapter has been published in Simossis VA, Heringa J (2004) The influence of gapped positions in multiple sequence alignments on secondary structure prediction methods. Comput Biol Chem 28:351-366.

4.1. ABSTRACT

All currently leading protein secondary structure prediction methods use a multiple protein sequence alignment to predict the secondary structure of the top sequence. In most of these methods, prior to prediction, alignment positions showing a gap in the top sequence are deleted, consequently leading to shrinking of the alignment and loss of position-specific information. In this paper we investigate the effect of this removal of information on secondary structure prediction accuracy. To this end, we have designed SymSSP, an algorithm that post-processes the predicted secondary structure of all sequences in a multiple sequence alignment by (i) making use of the alignment's evolutionary information and (ii) by re-introducing most of the information that would otherwise be lost. The post-processed information is then given to a new dynamic programming routine that produces an optimally segmented consensus secondary structure for each of the multiple alignment sequences. We have tested our method on the state-of-the-art secondary structure prediction methods PHD, PROFsec, SSPro2 and JNET using the HOMSTRAD database of reference alignments. Our consensus-deriving dynamic programming strategy is consistently better at improving the segmentation quality of the predictions compared to the commonly used majority voting technique. In addition, we have applied several weighting schemes from the literature to our novel consensus-deriving dynamic programming routine. Finally, we have investigated the level of noise introduced by prediction errors into the consensus and show that predictions of edges of helices and strands are half the time wrong for all four tested prediction methods.

4.2. INTRODUCTION

The use of multiple sequence alignments (MSAs) for secondary structure prediction has been one of the key innovations (Dickerson et al., 1976; Zvelebil et al., 1987; Rost and Sander, 1993) that have enabled prediction accuracy to reach its current average of 77%. By using a MSA instead of single sequences can improve the accuracy of a secondary structure prediction up to 8% (Levin et al., 1993). This, combined with the availability of increasingly larger protein sequence databases and more accurate search algorithms (Altschul et al., 1997; Altschul and Koonin, 1998), is the reason why

all current state-of-the-art secondary structure prediction methods use MSAs as input (see chapter 3). The explanation for this beneficial effect of MSAs is that they contain information about the evolutionary relationships of divergent proteins of the same

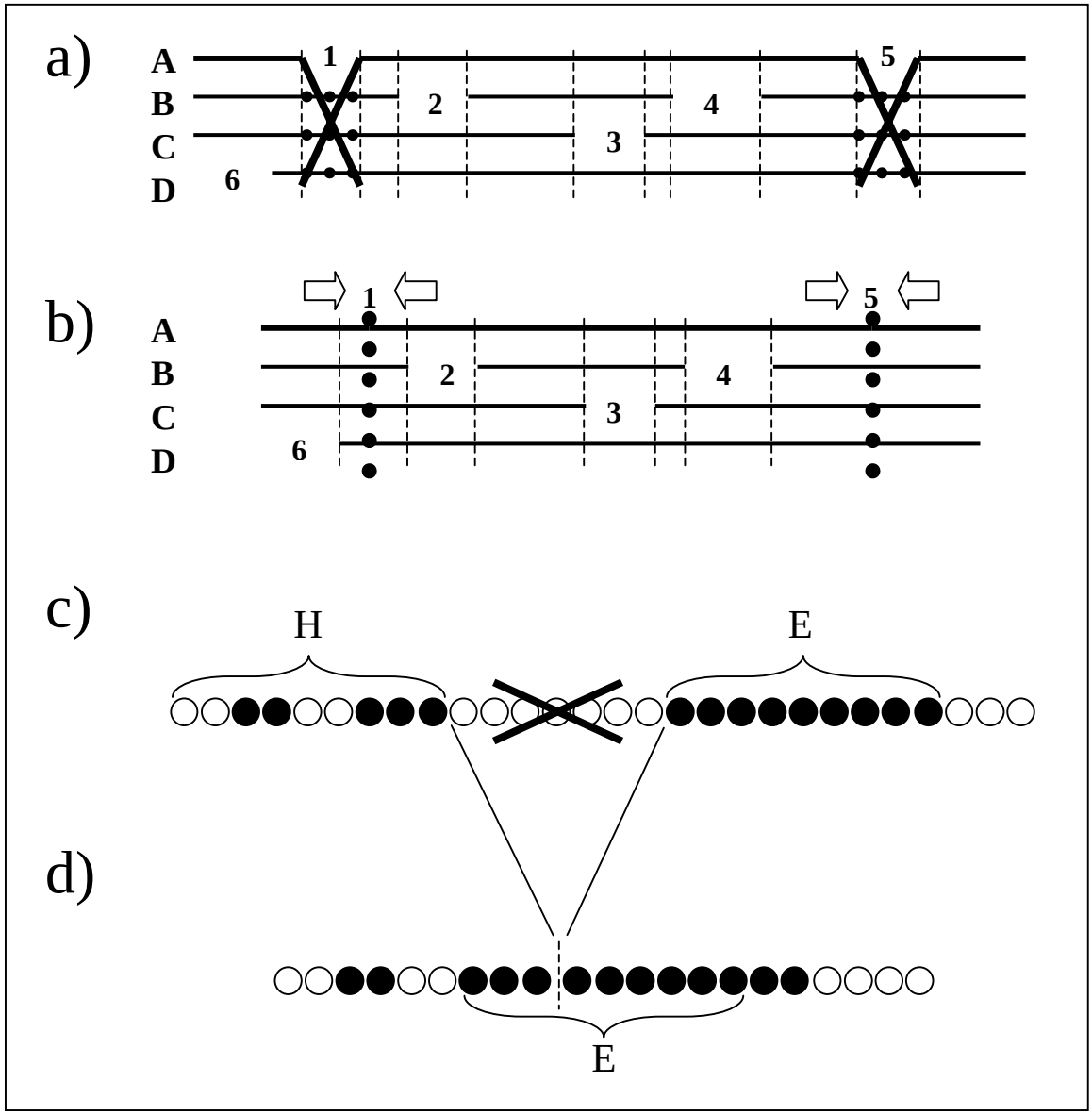


Figure 4.1. Diagrammatic representation of the pre-processing performed by secondary structure predictors, on input alignments and their possible effect. The circled numbers represent the gaps of the aligned protein sequences. (a) The gaps 1 and 5 of top sequence A cause the whole alignment position to be removed. The removed positions are covered by the large "X" 's. (b) Removal of the alignment positions shrinks the alignment and brings together the flanking regions of gaps 1 and 5, but also fuses the corresponding regions in the rest of the sequences as well. (c) A region with two distinct helix and strand hydrophobic (black) and hydrophilic (white) residue patterns are fused due to removal of the intervening residues. (d) The fusion of the two regions changes the pattern and leads to misprediction of part of the helical region as strand.

structural family. The position-specific information in a MSA about which residues vary within conserved secondary structure elements (SSEs) is crucial for accurate prediction.

Interestingly, state-of-the-art prediction methods such as PHD (Rost and Sander, 1993), PROFsec (Rost, personal communication), SSPro (Pollastri et al., 2002) and JNET (Cuff and Barton, 2000) remove whole alignment positions (columns), whose top position is a gap (Figure 4.1a), prior to prediction. As a result, the actual MSA that is used as input is missing information and at the same time regions of the MSA that are separate in sequence space fuse together (Figure 4.1b).

Considering the window sizes and correction filters used by the aforementioned prediction methods, it is conceivable that the removal of whole alignment positions (especially multiple consecutive positions) could lead to seriously erroneous predictions. For example, if two otherwise separate regions were fused together, this would lead to changes in the residue composition of that region and thus mispredictions (Figure 4.1c and 4.1d).

In this paper we will show that such pre-prediction processing often results in loss of evolutionary information; in contrast, when used it can significantly improve consistency and prediction accuracy. To this end, we have devised the SymSSP method (**Sym**metry-optimisation of **S**econdary **S**tructure **P**redictions), which based on the fact that protein structural elements are more conserved than their residue composition (Chothia and Lesk, 1986; Sander and Schneider, 1991; Rost, 1999) attempts to recover most of the missing information from the related sequences in the alignment and use it to improve the accuracy of the resulting prediction. First, the algorithm generates a library of iteratively derived secondary structures for each sequence in an input MSA. Ultimately, every sequence library contains a reference sequence and the derived secondary structures of the rest of the sequences in the MSA. For the purposes of this investigation we have tried to minimise the MSA-quality dependence of secondary structure prediction (Levin et al., 1993) by using the MSAs of the HOMSTRAD reference alignment database (Mizuguchi et al., 1998) (Jan 2003 release) as input for PHD, PROFsec, SSPro2 and JNET. In this way, we regard the MSA quality as optimal and focus on the quality of the secondary structure predictions alone. Finally, SymSSP

uses a novel weighted dynamic programming routine to produce an optimally segmented consensus secondary structure for each of the library reference sequences.

The use of dynamic programming for deriving an optimally segmented consensus has to our knowledge not been attempted yet and therefore we compared its performance to the so-called "majority voting" technique that is commonly used in several published consensus-deriving studies (Cuff and Barton, 1999; Albrecht et al., 2003; Ward et al., 2003). In addition, we have tested and combined many weighting scenarios from the literature in an attempt to derive a model for a priori weighting of the information. Finally, we have investigated the level of noise introduced to the consensus by the errors in predicted secondary structure elements and its implications on the resulting overall improvement.

4.3. MATERIALS AND METHODS

All described methods were run on the 72 dual Intel Pentium III 1.0GHz IBM cluster of the Computer Science department at the Vrije University Amsterdam. All secondary structure predictions were performed using a locally installed version of PHD, PROFsec, SSPro2 and JNET.

4.3.1 The SymSSP algorithm

The SymSSP algorithm proceeds in three main steps. Each step is described in detail in the sections below and applies to either of the four prediction methods used for this assessment:

a. Step 1: Creating secondary structure libraries

a) Getting a secondary structure prediction for all alignment sequences

We used PHD, PROFsec, SSPro2 and JNET to get a secondary structure prediction for all the sequences in the HOMSTRAD database. Since these methods only predict the secondary structure of the top sequence in an input alignment, we use a re-ordering routine to alternate the top sequence but retain the original alignment positions (Figure 4.2a and 4.2b).

b) Creating a secondary structure library for each sequence

As a result of pre-prediction processing, each alignment file has possibly had a portion removed for the prediction (Figure 4.1a). Note that these portions can be

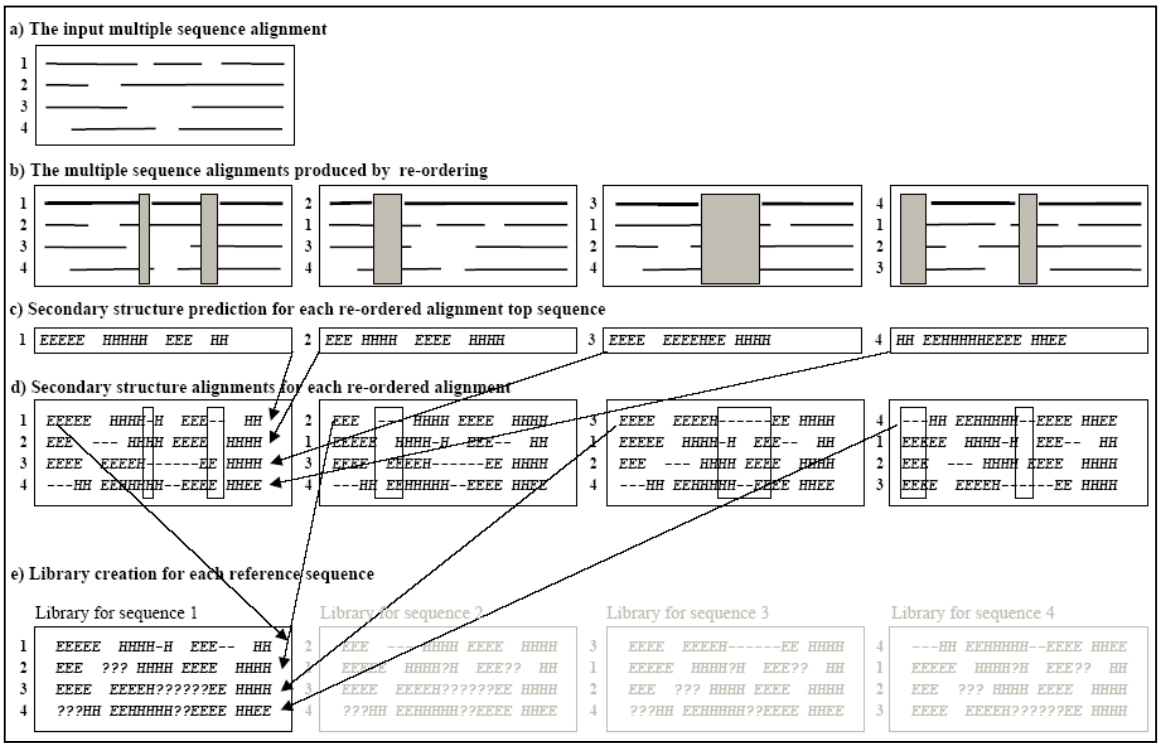


Figure 4.2. The creation of SymSSP libraries for sequence 1. (a) The original input alignment. (b) The re-ordering of the alignment to enable secondary structure prediction of all sequences; the grey boxes represent the regions that have been removed by pre-processing to give the predictions in (c). (c) The predicted secondary structures of the top sequences in the re-ordered alignments. (d) The secondary structure alignments using the predictions from all re-ordered alignments; the arrows show where each secondary structure in each alignment comes from. (e) Library creation from the secondary structure predictions of all sequences. The arrows show the origin of the predictions that make up the library. The actual predictions in the library are not all of sequence 1 but are modified versions of the other predictions that contain the information removed by the pre-processing on the sequence 1 alignment. The shaded grey boxes are the resulting libraries for sequences 2, 3 and 4.

different based on the positions of gaps in the input alignment and specifically on the placement of gaps in the top sequence.

We try to re-introduce the removed information by combining the predicted secondary structures of all sequences in the alignment into libraries. These libraries are produced by first combining the secondary structure predictions of all sequences in the alignment into a secondary structure alignment, using the original protein sequence alignment positions as a template (Figure 4.2c and 4.2d). We then consecutively regard each alignment sequence as a reference, mark the regions in the remaining predicted secondary structures that have no information about the reference sequence by representing them as a question mark (‘?’) and add them to

the library (Figure 4.2e). These '?' regions represent the positions that due to pre-processing would have been removed from the reference when each library member's secondary structure was predicted and therefore cannot contribute any information to the reference (boxes in Figure 4.2d). For example, in the library for sequence 1 in Figure 4.2e, the predicted secondary structure for sequence 2 will not contain information about the positions that were directly under its gaps and therefore those positions are marked with a '?'. The whole process is repeated for all sequences in the alignment.

b. Step 2: Information weighting and profile creation

To derive a consensus of the library information, we reduce each library to a secondary structure profile, which serves as a description of the SSE content of each library position in numerical values (Figure 4.3b). Each profile is optionally constructed using three combined weighting parameters, given in Figure 4.3a:

a) *Similarity weight (w_n):* The evolutionary distance of each library member to the reference sequence is optionally calculated using three different weighting schemes from the literature (P-Henikoff, BLOSUM and Lüthy). Each similarity weight was applied in three ways: global (the whole sequence), regional (each SSE separately) and positional (at each position). The principles and technical details of the different similarity weighting schemes tested are described below:

a. *No weights*: No evolutionary distances are taken into account. All library members are given a weight of 1.0 (w_n=1.0).

b. *P-Henikoff*: The philosophy of this scheme is that distant sequence predictions are up-weighted, while closely related ones are down-weighted to maximise the amount of information taken into consideration. Each library sequence SSE occurrence is given a weight according to a customised version of the Henikoff weighting scheme (Henikoff and Henikoff, 1994) (see equation 1):

$$w_n = H_{aa\,ss} = \frac{1}{n \times f(aa_{ss})} \quad (1)$$

where *ss* is the SSE type (one of helix, strand or coil (H, E or C)), aa_{ss} is residue *aa* in the sequence being weighted with predicted SSE type *ss*, H_{aass} is the Henikoff weight for residue *aa* with SSE type *ss*, *n* is the

number of dissimilar cases observed in that library position (column) and *f* is the frequency of aa_{ss} observed in that library position.

c. *BLOSUM*: In this scheme the weighting gives closely related sequences a higher weight. The similarity weight given to each matched residue pair is according to the BLOSUM62 matrix (Dayhoff et al., 1983).

d. *Lüthy*: This scheme is similar to the BLOSUM scheme, except that only the residue pairs that have identical SSEs are scored, according to the Lüthy set of secondary structure-specific exchange weight matrices for helix, strand and coil (Lüthy et al., 1994). This scheme differentiates the weighting of helix, strand and coil elements.

e. *Combined*: This scheme is a combination of the BLOSUM and Lüthy schemes. In this case, the similarity weight given to each matched residue pair is according to the exchange weights of the Lüthy matrices for the pairs with identical SSEs, while the rest use the BLOSUM62 matrix.

b) *Secondary structure-specific reliability weighting (w_{ss}):* PHD, PROFsec, SSPro2 and JNET provide reliability scores for each of their secondary structure-specific predictions with raw values ranging from 0 to 9 (0 is very low confidence). Reliability weighting uses these scores as weights, which were implemented into three types to represent global, regional and positional reliability:

$$w_{ss}(n,p) = \frac{R_{ss}(n,p)}{A} \times w_n \qquad (2)$$

where R_{ss} is the prediction method secondary structure-specific reliability value for position *p* in sequence *n*, *A* is the averaged scores of the region under test (i.e. a single position (*A*=1), SSE (*A*=average over element R_{ss}) or whole sequence (*A*=average over whole sequence)), and w_n is the weight given to that position according to the similarity weighting schemes described earlier. This weighting ensures that high reliability predictions score higher than less reliable ones.

c) *Consistency weighting (w_c)*: With this final weight, consistent positions are up-weighted with respect to less consistent ones, such as for example a position

with 60% helix and 40% strand. We converted the sum of the secondary structure-specific reliability weights (e.g. all w_{ss}'s for helix) for each alignment position to a *consistency weight* by normalising it with the sum of all secondary structure reliability weights for that alignment position:

$$w_c(ss, p) = \frac{\sum_{0}^{n-1} w_{ss}(n, p)}{\sum_{0}^{n-1} (w_H(n, p) + w_E(n, p) + w_C(n, p))} \quad (3)$$

where w_c is the consistency weight of secondary structure type *ss* (which can be H, E or C) at alignment position *p*.

c. Step 3: Deriving the consensus predictions

a) Dynamic Programming optimal segmentation

We use dynamic programming (DP) to find the optimum consensus from each library profile (Figure 4.3b). Once a profile is produced, it is used to fill a *LxL* search matrix, where *L* is the alignment length. Each column (*p*) represents the alignment position and each row (*l*) the length of a single secondary structure, the *window length*. Each cell in the matrix is filled with a window score (WS) that is calculated using the profile information (w_c). The window score is calculated as:

$$WS(ss, p, l) = \sum_{p}^{p+(l-1)} w_c(ss, p) \quad (4)$$

where *WS* is the window score of the secondary structure *ss* of length *l* starting at alignment position *p* and w_c is the consistency score calculated according to equation 3.

Window scores are produced for all three secondary structure possibilities and only the highest scoring one is used to fill the matrix. While the matrix is being filled, the path through the matrix is recorded in a separate *traceback array* (Figure 4.3b). At the end, the last cell of this array holds the score of the optimal path through the matrix. The information in the traceback array is then used to produce the final optimally segmented secondary structure consensus. The DP algorithm traverses the matrix without gap penalties.

b) Majority voting (MV) instead of DP

For comparison to our novel DP optimal segmentation strategy, we also applied

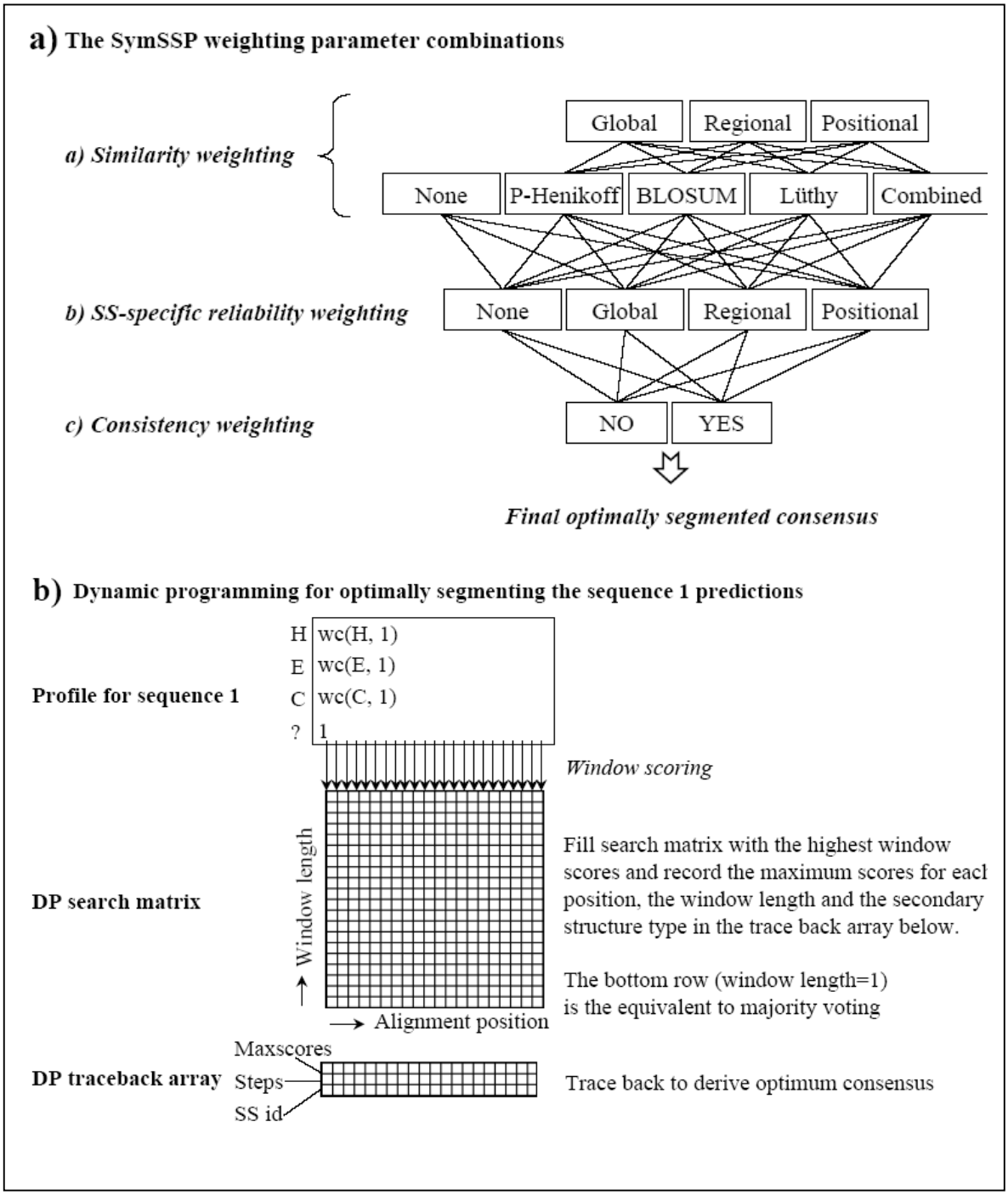


Figure 4.3. The SymSSP profile creation and DP optimal segmentation. (a) Flow chart representing all the weight combinations used in this investigation. The total number of parameter sets is 120 (i.e. 5×3×4×2). (b) The novel dynamic programming routine including window scoring. The bottom row (window length=1) of the DP search matrix is the equivalent of the majority voting technique. The left labels identify each structure in the algorithm and on the right we have provided a summary of the actions taken at each step.

the common "majority voting" technique to derive consensus predictions from the SymSSP library profiles. The "majority voting" technique was used in the initial

version of Jpred (Cuff et al., 1998) and since then has been commonly used in other investigations (Albrecht et al., 2003; McGuffin and Jones, 2003; Ward et al., 2003). For this strategy, we only consider one library profile column at a time and take the highest scoring secondary structure state as the state for that position (winner takes all), otherwise if no best scoring state can be derived the original prediction is used.

4.3.2 Assessment against DSSP

To assess the performance of our method in comparison to that of each prediction method, we used the DSSP method (Kabsch and Sander, 1983) to generate our standard of truth secondary structures from the PDB (Holm and Sander, 1996; Berman et al., 2000) files of the sequences in HOMSTRAD. We converted the DSSP 8-state secondary structure scheme (G, H, I=helix; E, B=strand; S, T, blank=coil) to the 3-state scheme (H=helix, E=strand, C=coil) to allow for accurate comparison with PHD, PROFsec, SSPro2 and JNET.

4.3.3 Edge effect calculation

Each position of a predicted helical or strand segment was considered with respect to its distance from the edge so that the outermost positions were given a position label of 1, the next one in 2 and so on, up (or down) to the midpoint. In case of even segments, the two innermost positions were given the same number. The total number of prediction errors with reference to DSSP at each segment position was recorded separately for helix and strand.

In addition, we performed the same calculation with respect to the DSSP segmentation (observed secondary structures). In this case, we recorded the errors in the predicted secondary structures based on the position labels given to the DSSP helix and strand segments.

4.3.4 Defining core and edge regions

The predictions generated from PHD, PROFsec, SSPro2 and JNET were separated into core and edge regions according to the edge effect calculations. Helix and strand edges were defined as two and one residues, respectively.

4.3.5 Prediction accuracy calculations

The quality of the PHD, PROFsec, SSPro2, JNET and SymSSP predictions were assessed using the Q3 and SOV (Zemla et al., 1999) scoring schemes. In addition, we calculated the Q3 scores separately for core and edge secondary structure regions, which was effected by disregarding core helix and strand regions for edge region evaluations, and *visa versa*. In all comparisons, the resulting scores were expressed both in terms of ΔQ3 and ΔSOV scores (SymSSP-method), where positive numbers denote an improvement.

4.4. RESULTS

The 779 alignments in HOMSTRAD were re-ordered as described in the methods section to produce a total of 2729 derivative alignments, each having a different top sequence. These alignments were then used as input to PHD, PROFsec,

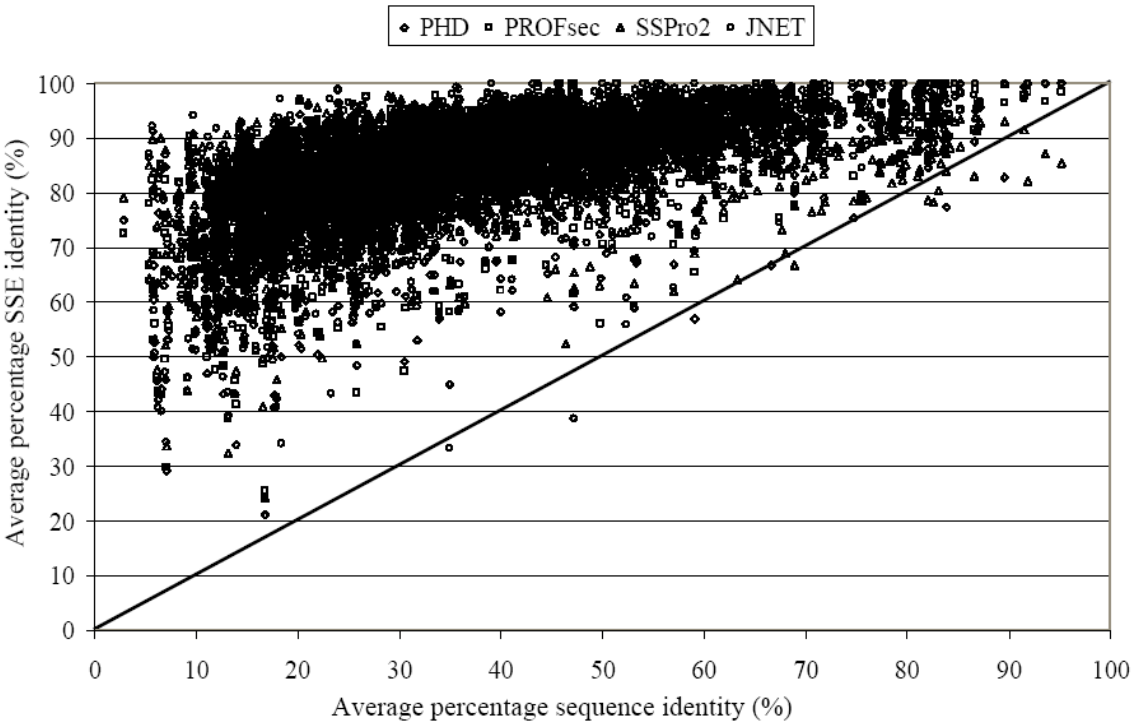


Figure 4.4. Correlation between the average sequence and predicted secondary structure identity of the library reference sequence and its corresponding library members over 2553 HOMSTRAD sequences for PHD, PROFsec, SSPro2 and JNET. Three PHD, 8 SSPro2 and 2 JNET cases show an exception to the rule and have higher sequence than structure identity, but this is a result of prediction error.

SSPro2 and JNET to predict the secondary structure for all 2729 top sequences.

SymSSP was run on all 2729 sequences in the HOMSTRAD database and 2553 sequences were found to have accurate corresponding DSSP entries. The sequences that were not included in the assessment contained amino acid differences between the HOMSTRAD source and the DSSP entries. If multiple model entries occurred in the DSSP files, we consistently selected the first one in the file. According to DSSP, from the 2553 sequences used, 170 were purely helical, 289 only contained strands, 6 contained neither helix nor strand elements and 2100 were mixed structures. Overall, 32% was helix, 22% was strand and 46% was coil.

4.4.1 PHD, PROFsec, SSPro2 and JNET delete alignment positions with gaps in top sequence

We compared the predictions made by PHD, PROFsec, SSPro2 and JNET on the original re-ordered alignments with those from a manually processed version of HOMSTRAD, where the alignment positions containing a gap in the top sequence were removed. We found that the predictions on both sets of alignments were identical, verifying that alignment pre-processing occurs prior to prediction (data not shown).

4.4.2 Sequence and structure conservation within libraries

In Figure 4.4, we have used the HOMSTRAD alignments to show how the average sequence identity between each reference sequence and its library members correlates with their corresponding average predicted secondary structure identity. The obviously higher conservation of secondary structure compared to the corresponding primary structure of the family clearly supports our strategy of using the related secondary structures as a means to re-introduce lost information.

4.4.3 Examples of SymSSP optimisation

In this section we use SymSSP-optimised examples of PHD, PROFsec, SSPro2 and JNET predictions to show how the re-introduced information corrects prediction errors and that the optimal segmentation protocol of SymSSP is more biologically correct than the common majority voting technique. In all examples the dynamic programming (DP) and majority voting (MV) strategies have been applied with no

weighting to show the baseline effects of the two consensus-deriving strategies (see Methods).

The SymSSP algorithm uses the related information from the rest of the family sequences and attempts to correct mispredictions that may have occurred due to pre-processing (removal of MSA positions with a gap in the top sequence). We illustrate the optimisation effect of SymSSP in Figure 4.5a using the *toxin* family alignment as an example. Interestingly, PHD, SSPro2 and JNET make a high number of mismatch errors in the same region (boxed), despite two of them being state-of-the-art methods (JNET, SSPro2). The sequences that do not have a gap region flanking that segment seem to not have the same errors. We have observed several cases where the regions flanking gaps have mismatch errors and involve all tested prediction methods.

This suggests that prediction pre-processing may have negative effects on the prediction quality of these regions, especially the edges close to the gaps. However, not every sequence with a flanking gap region shows errors so residue composition seems to be a crucial factor to whether it will affect the accuracy of the method or not. Nonetheless, the re-introduced information from the predicted structures of the un-processed sequences is used by SymSSP to correct most of the errors.

Why is DP more suitable than MV for processing the library information? The MV technique moves down the library sequences looking at segments of one position and returns a string of the best scoring positions. The DP approach tests all possible segment lengths at each position and returns the best scoring combination of segments. As a result, it keeps the consistent regions unchanged (like MV), but in addition optimises the edges where the highest variation occurs (edge effects). In the boxed segments in Figure 4.5b-d we show examples of how DP avoids prediction errors that MV is subjected to due to its ignorance of the surrounding elements, namely segment length errors (edge optimisation) and secondary structure type mismatches. However, even though the SymSSP algorithm is able to correct most prediction errors, it is limited by the accuracy of the predictions in the library and can only optimise the available information. As a result, corrections such as that marked in Figure 4.5b with a dashed box are not complete since they still count as wrong, but are a result of limited information. Similarly, consistent prediction errors like the one marked with a dotted box in Figure 4.5b sometimes cause DP to make a wrong correction compared to MV.

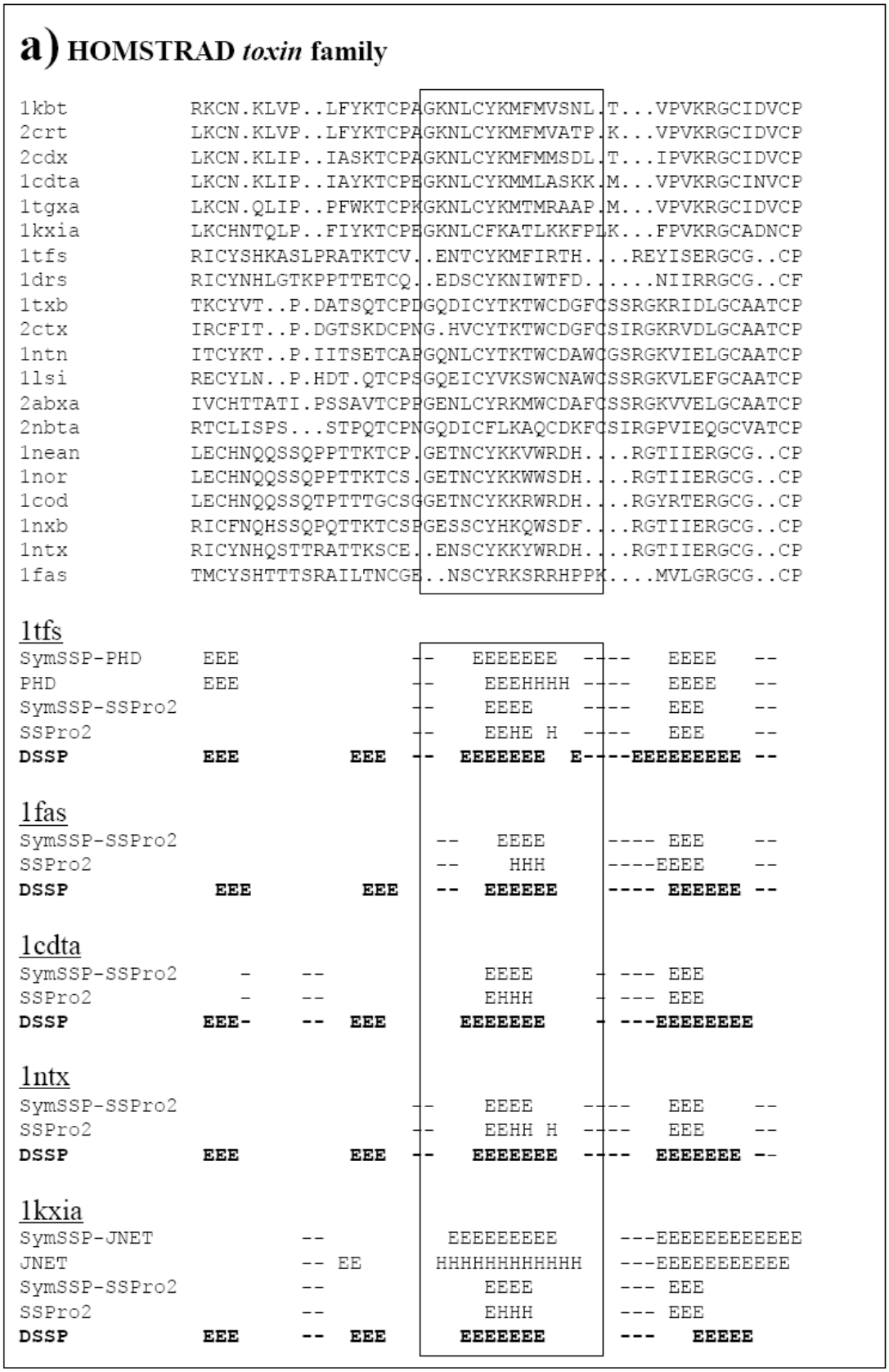


Figure 4.5. Next page

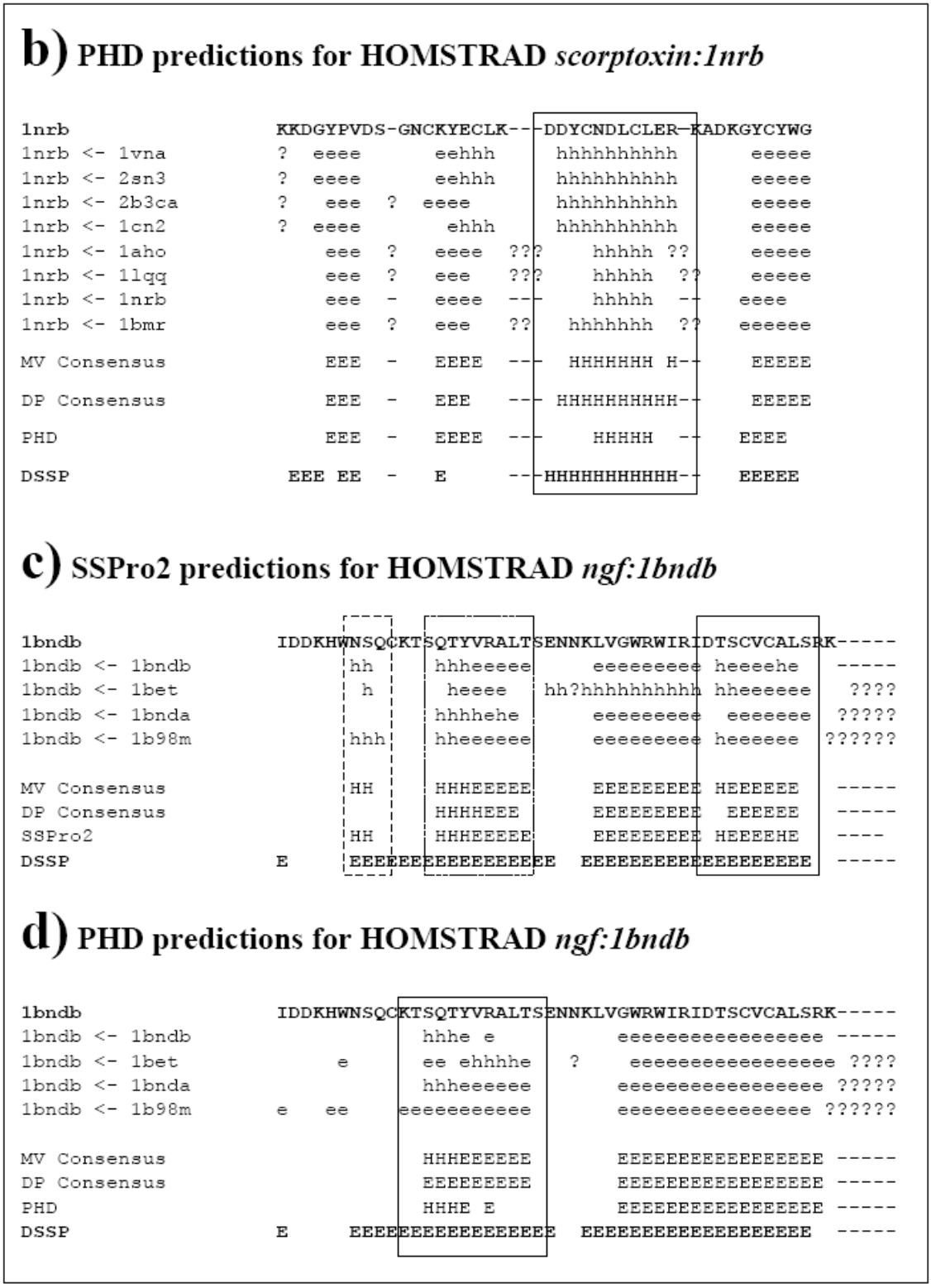


Figure 4.5. Examples of SymSSP optimisations. (a) The HOMSTRAD toxin family alignment and a list of some of the errors corrected by SymSSP using DP in PHD, SSPro2 and JNET. (b) the PHD prediction for sequence

Nonetheless, since most of the methods we are testing are state-of-the-art, such limitations are not frequent. From the four methods tested, PROFsec produced the most consistent libraries, whether wrong or not, and therefore the space for corrections was limited.

4.4.4 Overall correction of SymSSP

We calculated the overall effect of the SymSSP optimisations on the PHD, PROFsec, SSPro2 and JNET predictions derived from the HOMSTRAD alignments using the Q3 and SOV (Zemla et al., 1999) scoring measures (Table 4.1). The Q3 scoring strategy reflects how many single positions are correctly assigned, while the SOV score also takes into account the segmentation quality of the resulting prediction. The overall Q3 and SOV scores for PHD, PROFsec, SSPro2 and JNET on the 2553 HOMSTRAD sequences are listed in Table 4.1 for the original methods and for the methods in combination with three weighting modes of SymSSP (DP-weighted, DP-unweighted and MV). Table 4.1 shows that we attain a modest but consistent overall improvement for DP-weighted and DP-unweighted, but not for MV, which often shows negative overall results in the majority of cases.

4.4.5 Un-weighted SymSSP

The un-weighted SymSSP (Figure 4.3a - None, none, no) results show that without weighting the DP technique reduces the length of correctly predicted helices and strands compared to the MV strategy (Figure 4.6a – DP and MV columns). The overall ΔQ3 shows an apparent improvement, but this is mainly the result of over-predicting coil. This is due to the fact that the information in the un-weighted libraries is processed in the same way for all sequences and therefore the resulting optimised prediction is the same. Therefore, applying the same optimised prediction to all sequences in the alignment improves some and makes others worse. In HOMSTRAD there are 426 pair-wise alignments (34% of the sequences) where the DP strategy improves almost half but also makes an equal amount worse. The MV strategy, not having a majority option leaves all these sequences completely untouched. For the remaining HOMSTRAD alignments (>2 sequences), the DP strategy on average performs better than MV although the above-mentioned error is still present, albeit to a

much lesser extent than 50%. The "generalised" nature of the un-weighted SymSSP method was corrected by applying several weighting schemes to the library information and is discussed next.

In terms of segmentation quality, DP is clearly better than MV, which as explained earlier disregards segmentation. Despite the positional errors, it consistently produces an optimised segmentation leading to significant improvements in the SOV scores of SSPro2 and PHD (Table 4.1 and Figure 4.6b – DP and MV columns).

4.4.6 Weighted SymSSP

As explained in the Methods section, we tested several weighting strategies from the literature to augment SymSSP's ability to detect the correct signal from the library information depending on the reference sequence of each library. Contrary to the un-weighted approach, weighted optimisation will vary between libraries of the same alignment.

4.4.7 Similarity weighting improves handling of library information

The best scoring single set for both the DP and the MV strategies based on the resulting combination of ΔQ3 and ΔSOV scores was the BLOSUM global similarity, regional ss-specific reliability and consistency weighted set (Table 4.1 and Figure 4.6 – DP WEIGHTED). In all cases, the weighting enabled better selection of the library information and greatly repaired the errors made by the un-weighted SymSSP approach as shown by the ΔQ3 scores in Figure 4.7a. The most positively affected method was PHD. However, the extent of the error reduction was not enough to produce significant improvements in the Q3 scores compared to the original predictions. In terms of segmentation quality (SOV score), the weighting improved the helix and strand segmentation for all methods and particularly improved those of PHD and SSPro2. The latter, although being a top ranking method, made a large amount of length errors over our data that SymSSP could correct, leading to a 2% and 1% increase in its SOVH and SOVE scores, respectively, without affecting the SOVC score. In general, weighting forces strand and helix segments to become longer, leading to a smaller under-prediction of coil.

Table 4.1. Overall average Q3 and SOV scores for PHD, PROFsec, SSPro2, JNET and SymSSP using MV, DP and the BLOSUM DP weighting scheme over the 2553 HOMSTRAD sequences. Delta (Δ) is the difference between the SymSSP method and the ORIGINAL predictions (SymSSP-ORIGINAL). The Errsig is the significant difference as defined on the EVA server ($\sigma/\sqrt{N}$). The numbers in bold indicate significant improvements, i.e. Delta (Δ) scores greater than the Errsig of the method accuracy.

Method	**Q3**	**Q3H**	**Q3E**	**Q3C**	**SOV**	**SOVH**	**SOE**	**SOVC**
PHD	67.98%	63.11%	59.90%	70.34%	63.67%	62.49%	60.79%	61.84%
Errsig (significant difference)	0.29%	0.60%	0.54%	0.31%	0.34%	0.60%	0.56%	0.32%
SymSSP (DP-weighted)	68.23%	63.30%	60.31%	70.54%	64.31%	62.94%	61.91%	62.24%
Delta (Δ)	0.26%	0.20%	0.42%	0.21%	0.64%	0.45%	1.12%	0.40%
Errsig (significant difference)	0.29%	0.60%	0.54%	0.31%	0.34%	0.60%	0.56%	0.32%
SymSSP (DP)	68.31%	62.42%	59.32%	71.82%	63.94%	62.09%	61.39%	62.24%
Delta (Δ)	0.33%	-0.69%	-0.57%	1.48%	0.27%	-0.39%	0.60%	0.39%
Errsig (significant difference)	0.29%	0.60%	0.54%	0.30%	0.34%	0.60%	0.57%	0.32%
SymSSP (MV)	68.16%	63.15%	59.99%	70.53%	63.57%	62.36%	61.07%	61.61%
Delta (Δ)	0.19%	0.04%	0.10%	0.19%	-0.10%	-0.12%	0.28%	-0.23%
Errsig (significant difference)	0.29%	0.60%	0.54%	0.31%	0.34%	0.60%	0.56%	0.32%
PROFsec	69.77%	62.79%	61.62%	74.22%	66.17%	63.64%	63.49%	64.85%
Errsig (significant difference)	0.30%	0.60%	0.54%	0.30%	0.35%	0.61%	0.57%	0.33%
SymSSP (DP-weighted)	69.86%	62.92%	61.97%	74.11%	66.45%	64.02%	64.13%	64.93%
Delta (Δ)	0.09%	0.13%	0.35%	-0.12%	0.28%	0.38%	0.64%	0.08%
Errsig (significant difference)	0.30%	0.59%	0.53%	0.30%	0.35%	0.61%	0.57%	0.33%
SymSSP (DP)	69.84%	61.87%	60.79%	75.38%	66.01%	63.16%	63.38%	64.73%
Delta (Δ)	0.07%	-0.92%	-0.83%	1.16%	-0.16%	-0.48%	-0.11%	-0.12%
Errsig (significant difference)	0.30%	0.59%	0.54%	0.29%	0.35%	0.61%	0.57%	0.33%
SymSSP (MV)	69.80%	62.64%	61.60%	74.19%	65.98%	63.41%	63.67%	64.47%
Delta (Δ)	0.03%	-0.15%	-0.02%	-0.03%	-0.19%	-0.23%	0.17%	-0.38%
Errsig (significant difference)	0.30%	0.60%	0.54%	0.30%	0.35%	0.61%	0.57%	0.32%
SSPro2	69.72%	66.78%	57.77%	74.73%	64.39%	65.10%	60.91%	63.86%
Errsig (significant difference)	0.30%	0.55%	0.53%	0.30%	0.34%	0.57%	0.56%	0.32%
SymSSP (DP-weighted)	69.79%	66.66%	58.24%	74.63%	65.49%	66.65%	61.96%	64.02%
Delta (Δ)	0.06%	-0.12%	0.47%	-0.10%	1.10%	1.55%	1.04%	0.16%
Errsig (significant difference)	0.30%	0.56%	0.53%	0.30%	0.35%	0.58%	0.56%	0.32%
SymSSP (DP)	69.72%	65.74%	57.24%	75.72%	65.23%	66.08%	61.22%	64.07%
Delta (Δ)	0.00%	-1.04%	-0.54%	0.99%	0.84%	0.98%	0.30%	0.21%
Errsig (significant difference)	0.29%	0.56%	0.53%	0.29%	0.35%	0.58%	0.57%	0.32%
SymSSP (MV)	69.70%	66.63%	58.03%	74.56%	64.19%	64.80%	61.15%	63.61%
Delta (Δ)	-0.02%	-0.15%	0.25%	-0.16%	-0.20%	-0.29%	0.23%	-0.25%
Errsig (significant difference)	0.30%	0.55%	0.53%	0.30%	0.34%	0.57%	0.56%	0.32%
JNET	63.54%	63.98%	64.96%	59.47%	61.23%	61.03%	62.41%	57.93%
Errsig (significant difference)	0.28%	0.57%	0.48%	0.27%	0.32%	0.57%	0.52%	0.31%
SymSSP (DP-weighted)	63.64%	64.02%	64.85%	59.53%	61.25%	61.10%	62.49%	57.74%
Delta (Δ)	0.10%	0.03%	-0.11%	0.06%	0.01%	0.06%	0.08%	-0.19%
Errsig (significant difference)	0.28%	0.58%	0.48%	0.27%	0.32%	0.57%	0.53%	0.30%
SymSSP (DP)	63.77%	63.56%	64.36%	60.54%	61.29%	61.08%	62.49%	57.93%
Delta (Δ)	0.23%	-0.43%	-0.59%	1.07%	0.06%	0.05%	0.08%	0.01%
Errsig (significant difference)	0.28%	0.57%	0.48%	0.27%	0.32%	0.57%	0.53%	0.30%
SymSSP (MV)	63.61%	63.97%	64.90%	59.50%	60.94%	60.99%	62.53%	57.36%
Delta (Δ)	0.07%	-0.01%	-0.05%	0.04%	-0.30%	-0.04%	0.12%	-0.56%
Errsig (significant difference)	0.28%	0.58%	0.48%	0.27%	0.31%	0.57%	0.53%	0.30%

4.4.8 The effects of the secondary structure-specific reliability score adjustment

When the DP strategy was applied, the use of the ss-specific reliability weight increased both the overall Q3 and SOV scores of the resulting predictions in all cases, in comparison with not using this weighting at all. Overall, the most effective type appeared to be the regional ss-specific reliability weight. This is expected since the representation of each positional prediction reliability in relation to the whole SSE holding the position considered is more consistent with our DP strategy that takes into account the segmentation of the prediction.

For the MV strategy, the ss-specific reliability weight contribution was more dependent on the similarity-weighting strategy used. The positional ss-specific reliability weighting favoured the No-weighting and P-Henikoff similarity weighting schemes, while the regional approach favoured all the remaining similarity weighting schemes. However, the SOV scores consistently remained negatively affected for MV (Figure 4.6b).

4.4.9 The effects of the consistency score adjustment

Consistency scoring is a binary parameter (yes or no) of the applied weighting strategies (Figure 4.3). It performs a final adjustment to the weight of each sequence at every position of the profile according to the consistency of that position's predictions. In the case of the DP consensus-deriving strategy, its use benefited both the Q3 and SOV improvement scores in the majority of the weighting strategies, by an average of 0.1% and 0.05%, respectively. Conversely, it has no effect on the MV strategy performance (data not shown).

4.4.10 SymSSP alignment-specificity and weight-sensitivity

Since we have only applied single strategies to a large set of different alignment cases the overall performance scores for the applied weighting schemes do not reflect their alignment case-specificity. In other words, how much can be gained if we could apply the best weighting scenario for each individual case? To test the upper limits and the sensitivity of the weighting strategies we considered three scenarios: a) we kept one type of similarity weight constant and varied the ss-specific reliability and consistency

weighting; b) we kept one combination of ss-specific reliability and consistency weighting constant and varied the similarity weighting type and c) we varied all three weighting parameters. For each scenario we only recorded the best result for each sequence and found that scenarios a) and b) have the potential for an overall improvement in ΔQ3 and ΔSOV of over 1% and 2%, respectively, while scenario c) can improve the ΔQ3 and ΔSOV scores by 2% and 3%, respectively, for all tested prediction methods. In each scenario all sequence predictions either improved or were left unaltered. These results show that there is a strong case-specificity of the different weighted SymSSP methods and therefore, an adequate *a priori* detection scheme would allow SymSSP to fit the best weighting option to each case. We investigated the linear correlation of a set of 18 *ab initio* detectable sequence and alignment properties to the scores of each of the 120 parameter sets (Figure 4.3c), but the consistent observation was that the resulting ΔQ3 and ΔSOV scores were not obviously linked to any linear combination of the properties we considered (data not shown).

4.4.11 The level of noise introduced by prediction errors

As we have shown in the examples of SymSSP corrections, the optimised prediction quality is crucially dependent on the quality of the library information and the signal they provide. If the predicted structures in the libraries have high error rates, this will introduce noise into the data and the correct signal will become harder to detect. To determine the level of noise in the library data we investigated the positional prediction error levels of PHD, PROFsec, SSPro2 and JNET in helix and strand segments.

In order to have a common ground to compare the prediction errors of the methods, we used the DSSP-defined helix and strand segmentation. We recorded prediction errors as a function of distance (sequence positions) from the helix or strand segment edges up to the midpoint both from the N- and the C-terminal end (see methods section). Despite their difference in prediction accuracy, all four prediction methods go wrong almost half the time in the outermost edge positions of their predicted segments (Figure 4.7a and 4.7b). Errors are clearly highest in the two outermost positions of helical segments (2 at the N-terminus and 2 at the C-terminus)

and in the edge terminal positions of predicted strands (1 at the N-terminus and 1 at the C-terminus).

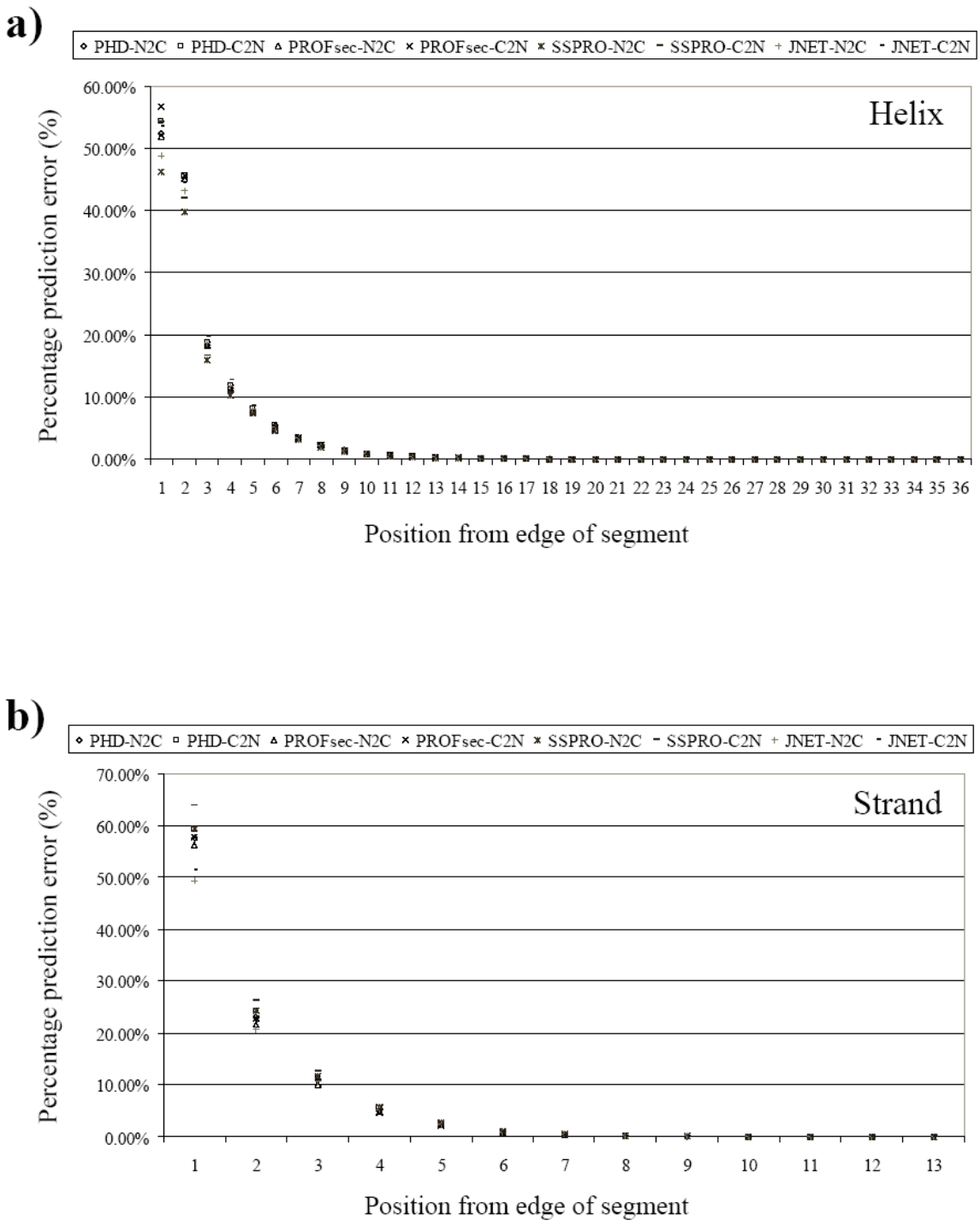
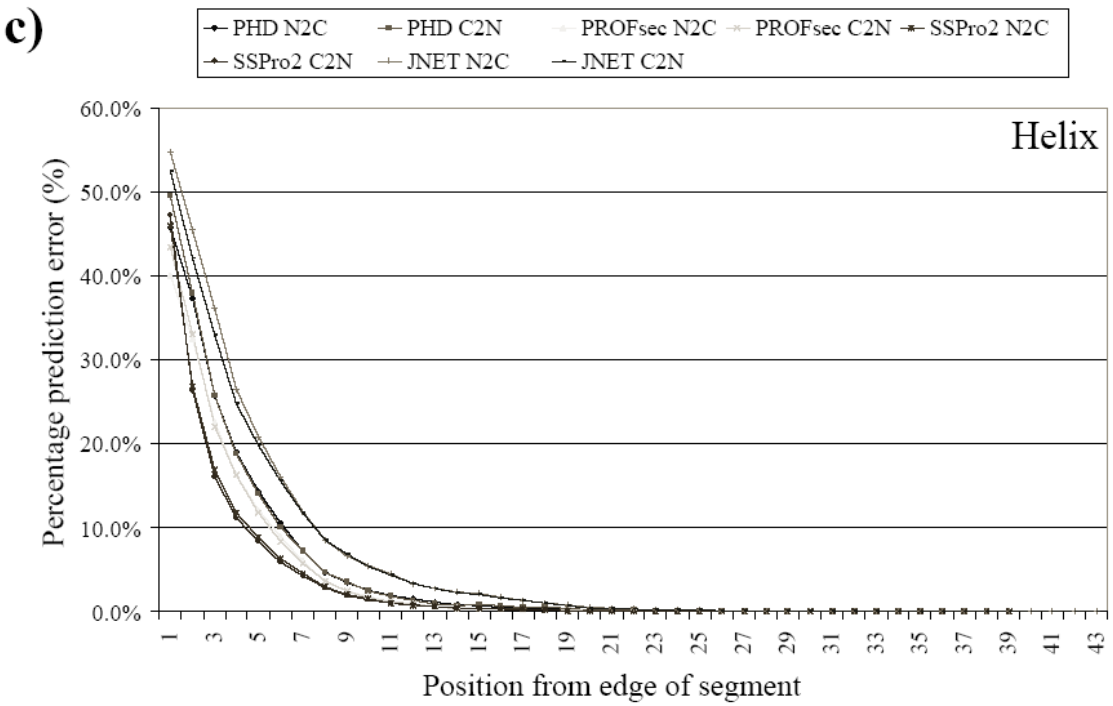


Figure 4.7. The calculated percentage error for each position from the edge to the midpoint for helix and strand predicted segments for PHD, PROFsec, SSPro2 and JNET according to (a-b) the DSSP segmentation and (c-d) the prediction segmentation. N2C and C2N signify the half-segment orientation from N-terminus to C-terminus and visa versa, respectively.



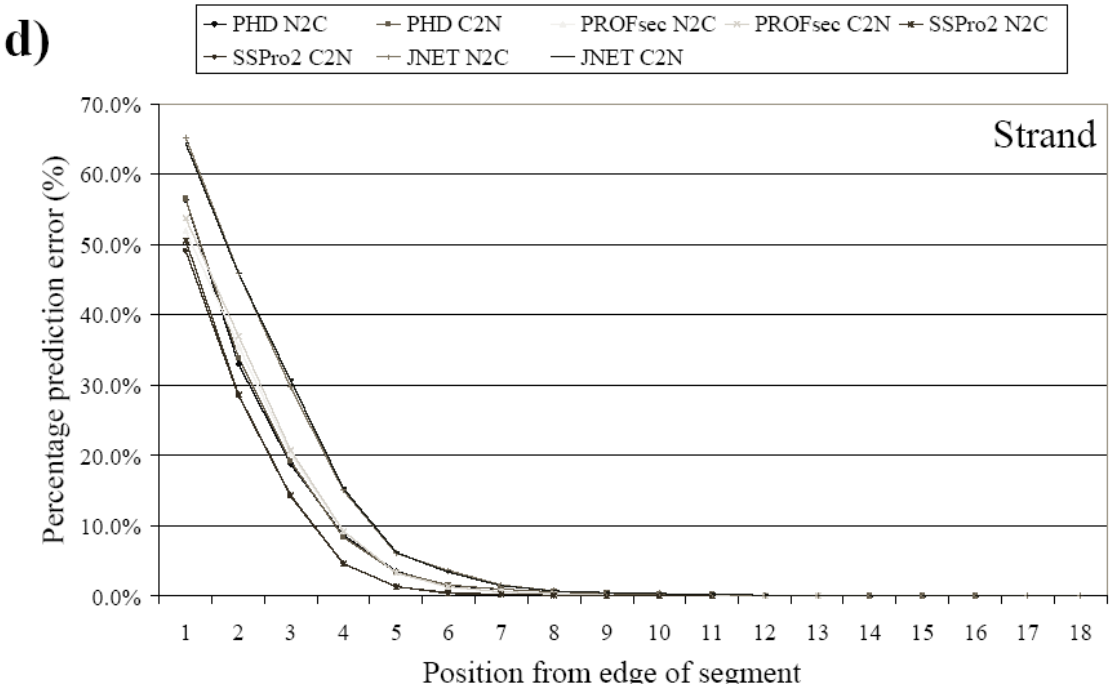


Figure 4.7. continued

In addition, the gradual drop in errors shows that prediction confidence increases towards the midpoint of the segments with most errors occurring in the six outer positions. This means that any information less than six positions-wide used for

predicting the structure of a single sequence position becomes increasingly unreliable. This six position window limit was also observed in the recent study by Crooks and Brenner (Crooks and Brenner, 2004). The above trend in position-specific error levels was also observed for the predicted segmentation of each method (Figure 4.7c and 4.7d).

The noisy input at the edges becomes a serious limitation to our optimal segmentation algorithm, since it mainly focuses on correcting edge prediction errors. To investigate the effects of the noise we assessed the Q3 accuracy of the SymSSP predictions with respect to PHD, PROFsec, SSPro2 and JNET in the core and edge regions with reference to the DSSP segmentation (Figure 4.8). In addition, we separated prediction errors into three types: i) mismatch (H→E and E→H), ii) under-prediction (H/E→C) and iii) over-prediction (C→H/E) (Table 4.2).

Table 4.2. The number of wrongly predicted positions in core and edge regions based on the DSSP segmentation for PHD, PROFsec, SSPro2, JNET (ORIGINAL) and the SymSSP approach using un-weighted DP (DP), un-weighted MV (MV) and BLOSUM weighted DP (DP-weighted), over the 2553 sequences in HOMSTRAD, separated into mismatches (MM), over-predictions (Llo) and under-predictions (Llu).

Method	**Core region**			**Edge region**		
PHD	*MM*	*Llo*	*Llu*	*MM*	*Llo*	*Llu*
ORIGINAL	13681	23559	78535	11927	40910	78535
SymSSP (DP-weighted)	13535	23351	78005	11806	40744	78005
SymSSP (DP)	13030	21926	81789	11288	39241	81789
SymSSP (MV)	13428	23250	78580	11742	40783	78580
PROFsec	*MM*	*Llo*	*Llu*	*MM*	*Llo*	*Llu*
ORIGINAL	11911	21427	79825	10320	35759	79825
SymSSP (DP-weighted)	11803	21570	79146	10305	35949	79146
SymSSP (DP)	11139	20059	83557	9807	34533	83557
SymSSP (MV)	11575	21305	80076	10190	35917	80076
SSPro2	*MM*	*Llo*	*Llu*	*MM*	*Llo*	*Llu*
ORIGINAL	10694	20628	81240	10678	35020	81240
SymSSP (DP-weighted)	10460	20567	81288	10413	35101	81288
SymSSP (DP)	9901	19222	85256	9923	33826	85256
SymSSP (MV)	10545	20515	81478	10450	35266	81478
JNET	*MM*	*Llo*	*Llu*	*MM*	*Llo*	*Llu*
ORIGINAL	20891	37326	59954	18128	54998	59954
SymSSP (DP-weighted)	20786	36990	60081	17973	55017	60081
SymSSP (DP)	20452	35773	62316	17683	53858	62316
SymSSP (MV)	20786	37037	60102	18001	54982	60102

Based on the ΔQ3 scores of the core and edge region positions it is clear that the un-weighted DP approach causes over-shortening of predicted segments, which leads to the increased level of under-predictions observed in Table 4.2. The resulting decrease in helix and strand Q3 accuracy is observed in both core and edge regions because shortening of the segments extends into the regions defined as core. Conversely, the weighted DP approach is able to improve the Q3 scores of both edge and core regions for almost all prediction methods. This improvement is a result of lower mismatch and under-prediction error levels observed for all prediction methods (Table 4.2).

4.5. DISCUSSION

We have investigated whether the common removal of alignment positions showing gaps in the top sequence prior to secondary structure prediction, as performed by most state-of-the-art prediction methods, affects the quality of the predictions. Our results show that re-incorporating information that would otherwise be lost by permuting the sequence order leads to an improvement in prediction quality. This is an important observation because from the methods we have tested, PROFsec (Rost, personal communication) and SSPro2 (Pollastri et al., 2002) are top-ranking prediction methods. In our investigation we have used these prediction methods without involving external search algorithms such as PSI-BLAST, which at the moment is used by most methods to gather position-specific information for their predictions. Instead we have given these methods structural alignments to work with, thus avoiding alignment quality-prediction accuracy dependencies.

In addition, the optimal segmentation strategy we have introduced in this study is to our knowledge the first attempt to use dynamic programming (DP) for deriving an optimally segmented consensus from a set of secondary structure predictions. We have shown that compared to the popular “majority voting” method used in Jpred (Cuff et al., 1998) and more recently in other studies (Albrecht et al., 2003; Ward et al., 2003), our strategy is able to prevent segmentation errors that otherwise occur due to the positional nature of majority voting and produce optimally segmented predictions. The weighted DP strategy was able to improve the SOV score of all tested methods and showed its highest improvement in the top-ranking method SSPro2 by increasing the

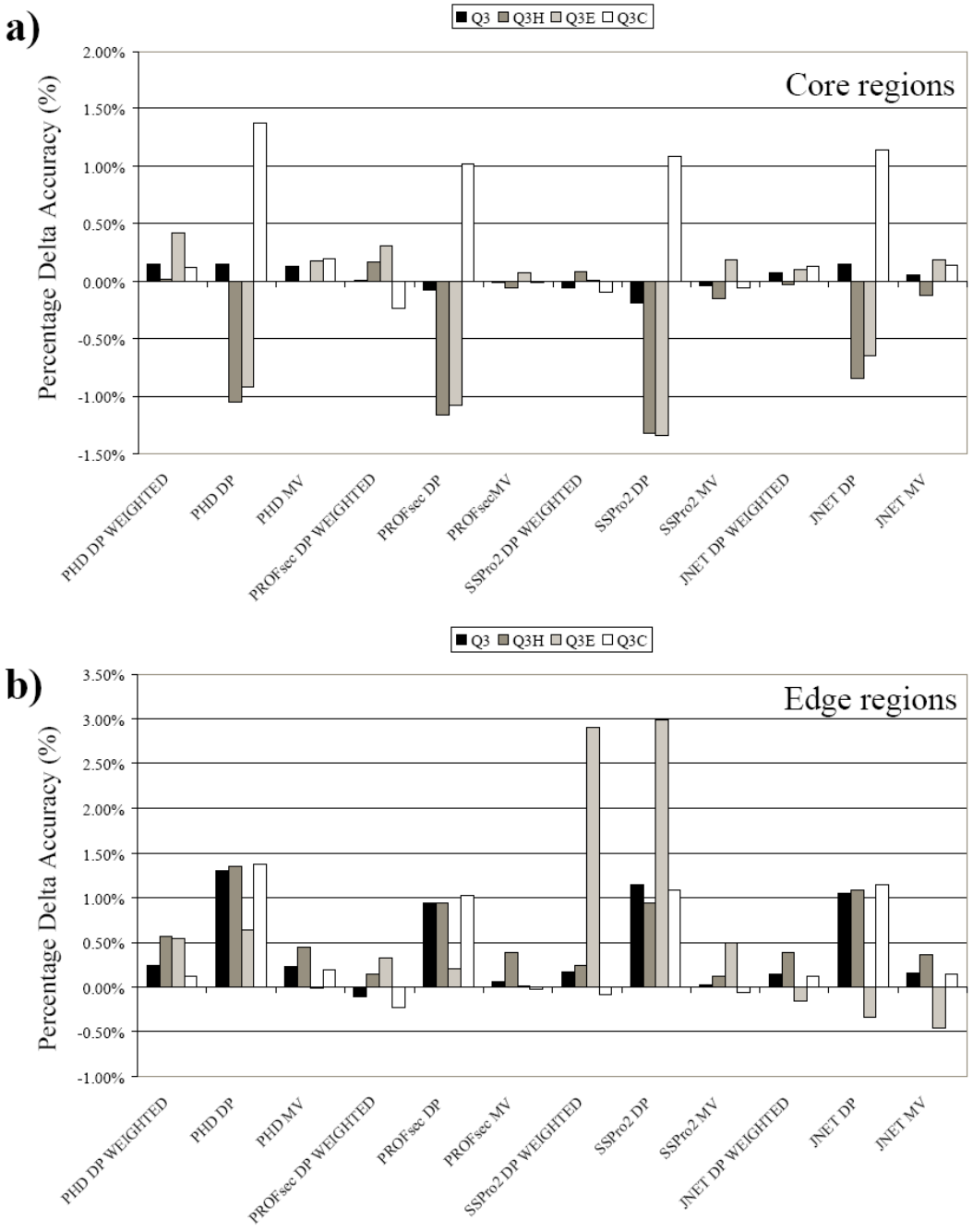


Figure 4.8. Average ΔQ3 scores of (a) the core regions and (b) the edge regions for SymSSP using the BLOSUM DP weighting scheme (DP WEIGHTED), un-weighted DP (DP) and un-weighted MV (MV) over the 2553 HOMSTRAD sequences.

overall SOV score by over 1%. To date, deriving a secondary structure consensus has mostly involved the use of predictions from separate prediction methods, rather than trying to optimise the output of one single method. This is an important difference because the variation in prediction accuracy of different methods on one sequence is far

less than the variation of one method on different related sequences (whether in an alignment or not). Consequently, the segmentation of this information becomes a more complex task. We have shown that incorporating weighting schemes to our DP strategy can improve overall prediction quality and has shown the capability of improving the prediction accuracy up to approximately 3%.

Finally, during this investigation we have observed that although the four prediction methods tested differ in overall accuracy up to over 6% on our test data sets (PROFsec and SSPro2 compared to JNET), the level of errors at the edge regions of predicted secondary structures is very similar. As a result, the noisy edge regions of predicted segments make it difficult for our DP strategy to detect the dominant signal for accurate consensus prediction and thus limit the extent of the overall Q3 improvement. Nonetheless, our optimal segmentation strategy consistently lowers mismatch and under-prediction errors in both the edge and the core regions.

The SymSSP method is freely available through an online interface at the Bioinformatics Section Server of the Vrije Universiteit Amsterdam (http://ibivu.cs.vu.nl/programs/symsspwww) where all weighting schemes and both MV and DP consensus-deriving methods can be used.

4.6. ACKNOWLEDGEMENTS

We would like to thank Dr. Jens Kleinjung for useful discussions and reading of the manuscript, Prof. Burkhard Rost for providing us with the PHD and PROFsec source codes, Dr. Gianluca Pollastri for the SSPro2 code, the JNET authors for making their method freely available online and the Vrije University Amsterdam for funding this project.

Chapter 5

A Simple and Fast Secondary Structure Prediction Method using Hidden Neural Networks

The content of this chapter has been published in Lin K[§], Simossis VA[§], Taylor WR, Heringa J (2005) A simple and fast secondary structure prediction method using hidden neural networks. Bioinformatics 21:152-159.

[§] Joint first authors

5.1. ABSTRACT

In this paper we present the secondary structure prediction method YASPIN that unlike the current state-of-the-art methods uses a single neural network to predict the secondary structure elements in a 7-state local structure scheme and then optimises the output using a Hidden Markov Model, which results in providing more information for the prediction. YASPIN was compared to the currently top-performing secondary structure prediction methods PHDpsi, PROFsec, SSPro2, JNET and PSIPRED. The overall prediction accuracy on the independent EVA5 sequence set is comparable to that of the top performers, according to the Q3, SOV and Matthew's correlations accuracy measures. YASPIN shows the highest accuracy in terms of Q3 and SOV score for strand and coil prediction. YASPIN is available online at the Centre for Integrative Bioinformatics website (http://ibivu.cs.vu.nl/programs/yaspinwww) at the Vrije University in Amsterdam and will soon be mirrored on the Mathematical Biology website (http://www.mathbio.nimr.mrc.ac.uk) at the NIMR in London.

5.2. INTRODUCTION

The field of secondary structure prediction has a history of over 40 years and a wide range of different models has been applied to tackle the problem (for reviews see (Heringa, 2000; Rost, 2001; Simossis and Heringa, 2004b)). Qian and Sejnowski introduced one of the earliest artificial neural network-based (NN-based) methods (Qian and Sejnowski, 1988). In this pioneering study, the supervised NN was trained using a non-redundant set of proteins with known structures and its secondary structure predictions were performed on single protein sequences. From the 1990s up to present times, secondary structure prediction accuracy has improved to over 70% by incorporating the evolutionary information found in multiple sequence alignments. Among the most successful methods to date, PHD (Rost and Sander, 1993), PHDpsi (Przybylski and Rost, 2002), PROFsec (Rost, personal communication), SSPro2 (Pollastri et al., 2002), JNET (Cuff and Barton, 2000) and PSIPRED (Jones, 1999) employ various types of neural networks to perform predictions using multiple sequence alignments (MSAs) of homologous sequences. However, the improvement in secondary structure prediction accuracy through use of MSAs is also directly connected

to database size and search accuracy (Przybylski and Rost, 2002). As a result, all currently top-performing methods, including the ones mentioned above, employ the iterative databank-searching tool PSI-BLAST (Jones, 1999; Pollastri et al., 2002) to select homologous sequences for predicting secondary structure. The PSI-BLAST output is used in several ways, but most methods use either the final alignments or the resulting position-specific scoring matrices (PSSMs). In the currently reigning prediction method PSIPRED the PSSM from PSI-BLAST is directly taken and used for prediction. The prediction performances of these programs have been extensively documented in various assessments (Jones, 1999; Jones and Swindells, 2002; Albrecht et al., 2003; Koh et al., 2003; McGuffin and Jones, 2003).

Many of the current NN-based methods use feed-forward multi-layer perceptron networks, which are trained with the back-propagation algorithm (Bishop, 1995). The first layer network predicts the secondary structure of the central residue of a preset window size, according to a PSSM and/or another form of multiple sequence alignment encoding. This is called the sequence-to-structure network. The second layer network, called the structure-to-structure network, filters the outputs from the first one and produces the final prediction results. Additional layers of networks or other decision-making models can further complement each of these network layers. For example, in the Prof method (Ouali and King, 2000), the final prediction results were obtained using four layers, a large number of NNs, combined with linear discrimination of multiple cascaded classifiers.

Since the Qian and Sejnowski 1988 paper, the number and complexity of NNs used in secondary structure prediction has increased dramatically. In YASPIN we apply a single NN on the same prediction task instead of employing complex multi-layered networks of NNs. However, the problem with using a single NN is that the prediction results are often “broken” secondary structures, even elements of only one residue. This is not desirable as most observed secondary structures are composed of more than three residues. A common way to overcome this problem is to filter the predicted secondary structure elements from the NN by using additional NNs. In YASPIN we apply a Hidden Markov Model (HMM). The forward and backward algorithms of the HMM are also used to assign the confidence for each prediction (prediction reliability scores). Finally, the prediction results are converted into 3-state secondary structure predictions

('H'-helix, 'E'-strand, '-'-other). YASPIN can be trained in a few days and can process a prediction in a few seconds.

5.3. MATERIALS AND METHODS

5.3.1 The algorithm

YASPIN is a Hidden Neural Network (HNN) secondary structure prediction method. It uses a feed-forward perceptron network with one hidden layer to predict the secondary structure elements from the sequence. Then, these predictions are filtered by a Hidden Markov Model (HMM).

The YASPIN neural network (NN) uses the soft-max transition function (Bishop, 1995) with a window of 15 residues. For each residue in that window, 20 units are used for the scores in the PSSM and 1 unit is used to mark where the window spans termini of protein chains. In total, the input layer has 315 units (21 x 15). For the hidden layer we use 15 units. The output layer has 7 units, corresponding to 7 local structure states: helix beginning (Hb), helix (H), helix end (He), strand beginning (Eb), strand (E), strand end (Ee) and coil (C). The beginnings and ends of the helix and strand elements we refer to are single residue positions.

The 7-state output of the NN is then passed through a Hidden Markov Model (HMM), which uses the Viterbi algorithm (Viterbi, 1967; Durbin, 1998) to optimally segment the 7-state predictions. The HMM defines the transition probabilities between the 7 local structure states (see Figure 5.1). The final output is a 3-state secondary structure prediction ('H' for helix, 'E' for strand and '-' for coil).

5.3.2 Testing and training datasets

YASPIN was trained and tested using the SCOP1.65 database (Murzin et al., 1995; Hubbard et al., 1998). The test and training sets were built using the PDB25 set (25% maximum sequence identity) grouped together by ASTRAL (Brenner et al., 2000). Before using the PDB25 dataset we removed all transmembrane entries (SCOP class f) resulting in a non-redundant set of 4256 proteins with known structures. The test set was extracted before training by random selection from the complete PDB25 set at a ratio of approximately 1:8. The 535 sequences selected for the test set were a) at

most 25% identical to the training set due to the nature of the PDB25 dataset and b) were not part of the same superfamily as any of the remaining 3721 sequences of the training set, according to the SCOP superfamily definitions.

In addition, in order to make a more accurate comparison between all methods, including YASPIN, we further benchmarked all methods on the independent "common_set 5" dataset (10-2002) from EVA (Koh et al., 2003). To this end, we removed any sequences found in the EVA5 sequence set from the YASPIN training set. The final YASPIN training set contained 3553 sequences with known structures.

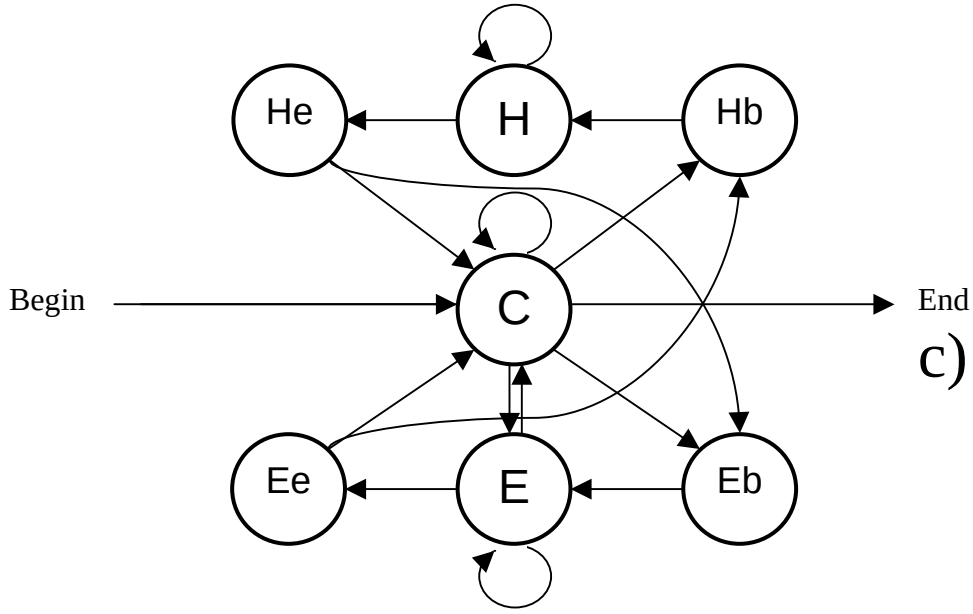


Figure 5.1. The Hidden Neural Net (HNN) state diagram. The arrows represent the allowed transitions in the HNN. H, E and C represent α-helix, β-strand and coil, respectively. The labels 'b' and 'e' indicate beginnings and ends of secondary structures.

5.3.3 NN and HMM training

To train the YASPIN NN, we used the on-line back-propagation algorithm and six-fold cross-validation (Bishop, 1995). In a single training iteration, each of the six subsets was successively kept apart for testing, while the remaining five were used to train the network. At the end of each training iteration the average prediction error of the networks over all six test subsets was recorded and when the average prediction error started to increase, the training was stopped. We used a momentum term of 0.5

and a learning rate of 0.0001.

The reference secondary structure states used to train the HMM were obtained using DSSP (Kabsch and Sander, 1983). The DSSP 8-state secondary structure representation (H, G, E, B, I, S, T, -) was grouped according to the 3-state scheme

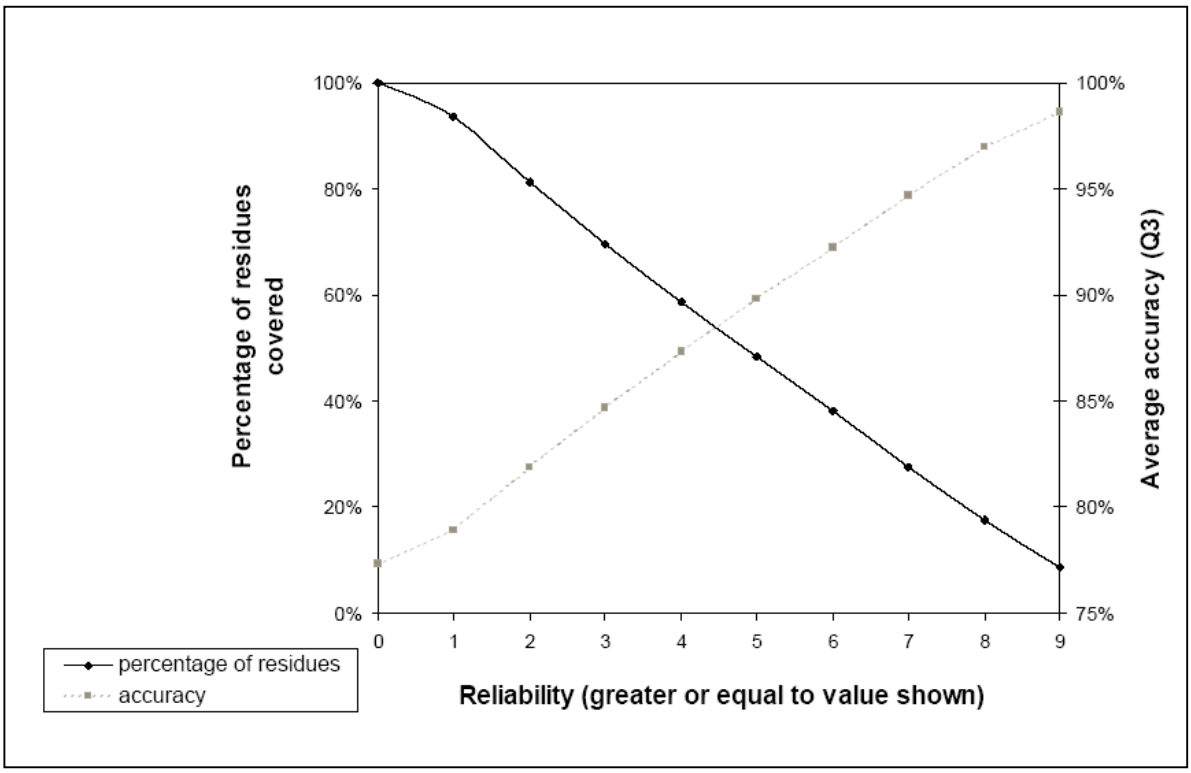


Figure 5.2. Average secondary structure prediction accuracy (Q3), and percentage of residues against cumulative reliability index from the YASPIN method. For example, for residues with reliability index of ≥ 6, the average accuracy is 92%, and percentage of residues with this index is 38%.

proposed by Rost and Sander (1993), i.e. H and G were considered as helix (H), E and B as strand (E), and all others as coil (C). These 3-state definitions were later converted to our 7-state local structure scheme (see Figure 5.1). The transition probabilities of the HMM were estimated using the training set.

5.3.4 Reliability scores

The YASPIN prediction algorithm provides four different position-specific prediction confidence scores (reliability scores). These scores are generated based on the NN predicted probabilities of each residue being in one of the defined 7 states. The first three scores are secondary structure-specific scores, representing helix, strand and

coil prediction confidence and are generated as the sums of the probabilities of each respective secondary structure type. For example, let a residue X have a probability of being in any of the 7 states. Its helix confidence score would be the sum of the Hb, H and He scores for that position. These three scores are normalised to always add up to 9.

The fourth score is the position-specific prediction confidence number, which represents the score of the state the Viterbi algorithm has chosen in its optimal segmentation path. All four scores are estimated using the HMM forward and backward algorithms.

5.3.5 PSSMs

All sequences in the test set were sequentially used as queries in a PSI-BLAST search against the non-redundant database (NR). All involved secondary structure prediction methods were tested on the same PSI-BLAST results to make the comparison as unbiased as possible. The search parameters were set to satisfy the formatting and output needs of all the involved methods according to the suggestions of their corresponding authors. We used a cut-off of 0.001 (-h 0.001) as suggested by the PSIPRED parameter settings, a maximum of 3 iterations (-j 3), output formatting of type 6 which is needed by JNET and finally also generated PSSM and Check files for each sequence. The actual command line was “blastpgp -i [query sequence] -h 0.001 -m 6 -j 3 -d nr -Q [PSSM] -C [CHECKFILE] > [BLAST OUTPUT]”.

5.3.6 Benchmarking

Benchmarking of YASPIN was performed using locally installed versions of the PHDpsi, PROFsec, SSPro2, JNET and PSIPRED programs. PHDpsi and PROFsec predictions were performed using the extracted alignments of the PSI-BLAST run. JNET was run using the extracted PSI-BLAST alignments, the PSI-BLAST PSSM files and the generated frequency profile files, according to the authors’ instructions. The HMM profiles were only included in the prediction when available.

YASPIN’s prediction accuracy was compared to that of PHDpsi, PROFsec, SSPro2, JNET and PSIPRED, using the corresponding DSSP-derived secondary structures as a standard of truth. The translation from 8-state to 3-state secondary

structure classification was performed according to the EVA (Koh et al., 2003) conversion scheme. The prediction accuracy of all methods was measured using the standard formulas for the Q3, SOV (Zemla et al., 1999) and Matthew's correlation coefficients (for review see (Simossis and Heringa, 2004b)) as given on the EVA server (Koh et al., 2003).

5.3.7 Calculating prediction errors

We separated the prediction errors for helix and strand into four classes, in accordance to the classification used in McGuffin and Jones (McGuffin and Jones, 2003): a) wrong prediction (w), b) overprediction (o), c) under-prediction (u) and d) length (l) errors. The length errors were also recorded separately as over and under-predictions for comparison purposes between the methods. The four error types are illustrated below for clarity.

```
AA      MDYFTLFGLPARYQLDTQALSLRFQQLAAVQTINQ…
SS             HHHH    EEEEE HHH HHHHH     …
DSSP     HHH   HHHH HHHHHHHHHH             …
Errors   uuu       uuuwwwwwl ll ooooo     …
```

5.4. RESULTS

YASPIN was trained on a non-redundant set of 3553 proteins with known structure from the PDB25 SCOP1.65 database. Its performance was tested using 535 proteins with known structure from the PDB25 dataset that were neither present in the training set nor were part of the same SCOP-defined superfamily as any structure in the training set.

The PDB25 test set was also used to compare YASPIN to current top-performing methods PHDpsi, PROFsec, SSPro2, JNET and PSIPRED. From the 535 sequences in the test set, 409 were found to be common to all methods, i.e. all methods returned a prediction for these proteins. This comparison was relatively unfair for YASPIN since many of these state-of-the-art methods have used sequences from this test set for their training. Nonetheless, the Q3 and SOV score results in Table1a show that YASPIN is the best in strand prediction and also outperforms most methods in

helix prediction except SSPro2 and PSIPRED, which are clearly superior to YASPIN in that respect.

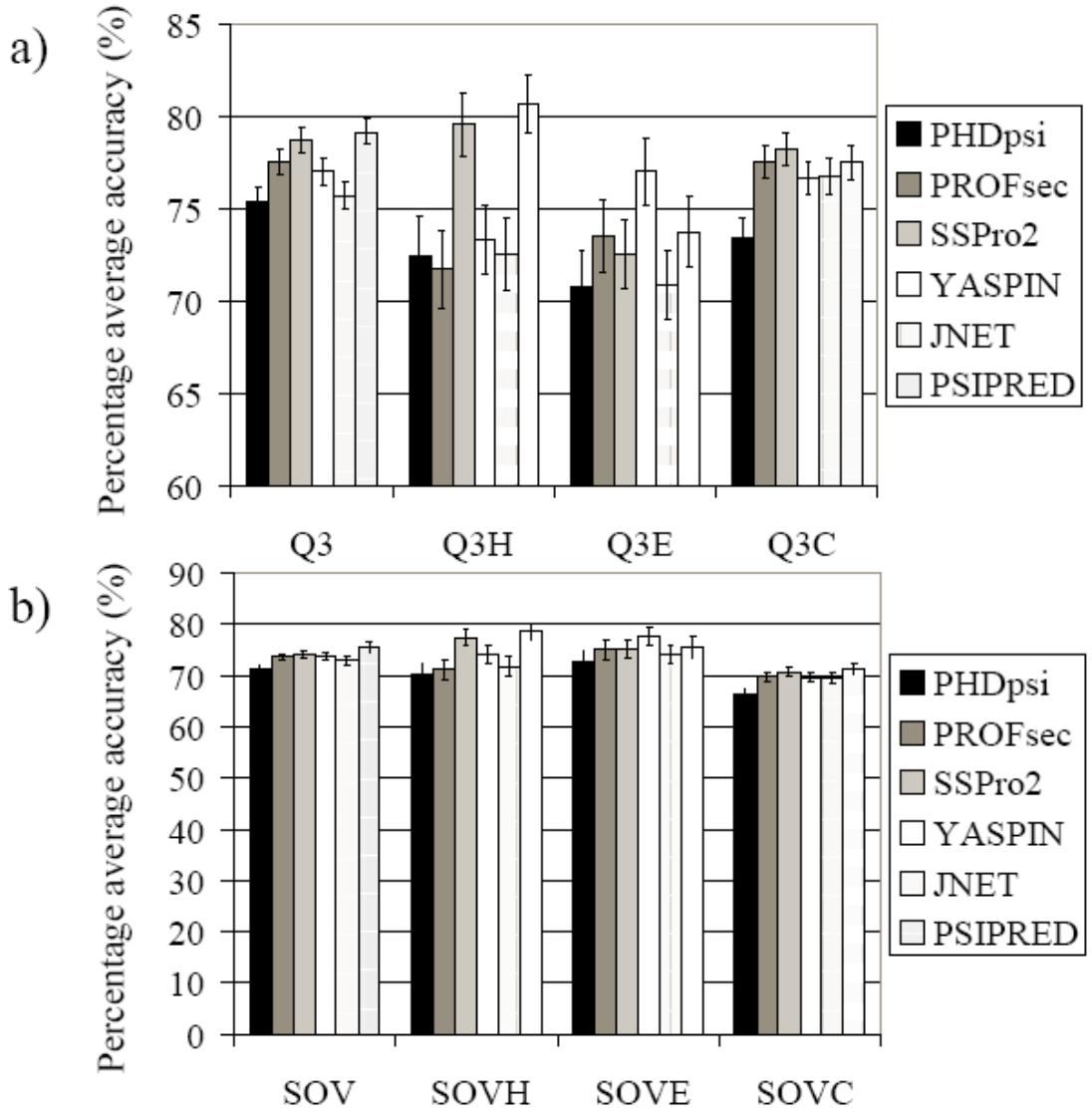


Figure 5.3. The (a) Q3 and (b) SOV scores for PHDpsi, PROFsec, SSPro2, YASPIN, JNET and PSIPRED on the independent EVA5 common dataset (188 sequences). Q3H/E/C and SOVH/E/C values are the specific Q3 and SOV scores of the predicted helical, strand and coil regions, respectively.

In addition, these methods were also benchmarked against the independent EVA5 sequence set (cumulative 10/2002). Since the EVA5 sequences were removed from the YASPIN training set (see methods) and all the other methods did not include these cases in their training, this dataset allows us to accurately compare YASPIN to these methods as well as the methods between themselves. From the 217 sequences in the EVA5 test set, 188 were found to be common to all methods. The prediction

accuracies were assessed in three ways: a) the three-state per-residue prediction accuracy measure (Q3) (Figure 5.3a), b) the segment overlap measure (SOV) (Figure 5.3b), both calculated using the SOV software (Zemla et al., 1999) and c) the Matthew's correlation coefficients (MCC's) (Table 5.2b).

Table 5.1. The average Q3 and SOV scores for the predictions of (a) 409 PDB25 common sequences from the testing set and (b) 188 common sequences from the EVA5 set, with respect to the DSSP reference databases. Q3H/E/C and SOVH/E/C values are the specific Q3 and SOV scores of the predicted helical, strand and coil regions, respectively. Errsig is the significant difference margin for each score and is defined as the standard deviation (σ) over the square root of the number of proteins ($\sqrt{N}$). All values are averaged over all α, β, $\alpha+\beta$ and α/β proteins.

a)

PDB25	**Q3**	**Q3H**	**Q3E**	**Q3C**	**SOV**	**SOVH**	**SOVE**	**SOVC**
PSIPRED	67.63	69.36	55.17	71.15	63.14	67.39	57.36	61.99
Errsig	0.96	1.41	1.61	0.94	1.08	1.47	1.70	1.01
SSPRO2	67.39	67.22	52.91	72.78	62.33	65.30	55.75	62.28
Errsig	0.94	1.47	1.59	0.93	1.06	1.53	1.67	0.96
PROFsec	66.64	62.92	55.91	71.89	62.70	63.02	57.92	62.24
Errsig	0.91	1.51	1.56	0.91	1.04	1.58	1.64	0.95
YASPIN	66.41	64.34	58.40	69.92	62.15	63.82	58.87	60.60
Errsig	0.95	1.49	1.60	0.94	1.05	1.54	1.66	0.97
JNET	65.41	61.84	54.58	70.85	61.02	60.72	57.10	60.59
Errsig	0.90	1.51	1.56	0.94	1.01	1.57	1.64	0.93
PHDpsi	65.03	63.49	54.92	67.76	60.27	61.74	55.93	59.19
Errsig	0.90	1.51	1.57	0.96	0.99	1.53	1.61	0.93

b)

EVA5	**Q3**	**Q3H**	**Q3E**	**Q3C**	**SOV**	**SOVH**	**SOVE**	**SOVC**
PSIPRED	79.20	80.69	73.77	77.56	75.62	78.68	75.53	71.29
Errsig	0.68	1.60	1.93	0.93	1.09	1.74	2.02	1.14
SSPRO2	78.77	79.58	72.53	78.23	74.17	77.39	75.20	70.75
Errsig	0.71	1.71	1.86	0.92	1.13	1.85	1.98	1.11
PROFsec	77.56	71.73	73.52	77.56	73.61	71.17	75.18	69.74
Errsig	0.70	2.10	1.95	0.91	1.08	2.18	2.04	1.13
YASPIN	77.06	73.35	77.05	76.72	73.88	74.19	77.64	69.67
Errsig	0.74	1.83	1.87	0.90	1.11	1.90	1.95	1.19
JNET	75.72	72.55	70.89	76.75	72.94	71.78	74.21	69.54
Errsig	0.73	1.97	1.90	1.01	1.07	2.03	1.96	1.16
PHDpsi	75.44	72.47	70.79	73.39	71.19	70.19	72.78	66.23
Errsig	0.75	2.12	1.98	1.07	1.08	2.15	2.03	1.16

Table 5.2. a) The different error types and b) the Matthew's correlation coefficients for the YASPIN predictions in comparison to the other methods on the 409 PDB25 and 188 EVA5 common test sets. (H/EW: wrong prediction (H→E, E→ H), H/EO: helix or strand structure over-predicted, H/EU: helix or strand structure under-prediction and H/EL: helix or strand structure length error).

a)

	PDB25								EVA5							
Methods	**HW**	**HO**	**HU**	**HL**	**EW**	**EO**	**EU**	**EL**	**HW**	**HO**	**HU**	**HL**	**EW**	**EO**	**EU**	**EL**
PHDpsi	1670	336	2153	9165	2281	1193	2277	5594	369	196	888	2843	362	283	751	2330
PROFsec	1548	344	2241	8506	1964	1043	2347	5472	320	210	873	2529	253	259	732	2111
SSPro2	1181	793	1701	8495	2150	896	2378	5051	236	381	608	2457	340	206	698	1959
YASPIN	2037	441	2087	8332	1908	1058	2165	5857	515	242	827	2316	186	287	712	2079
JNET	1569	368	2406	8887	2249	1052	2204	5453	397	168	940	2732	527	263	634	2160
PSIPRED	1297	604	1757	8311	2182	856	2221	4992	225	291	650	2339	343	167	725	1879

b)

PDB25	**MCC**	**MCC_H**	**MCC_E**	**MCC_C**
PHDpsi	0.45	0.52	0.43	0.40
PROFsec	0.48	0.54	0.46	0.43
SSPro2	0.49	0.56	0.47	0.45
YASPIN	0.47	0.54	0.45	0.42
JNET	0.46	0.52	0.45	0.42
PSIPRED	0.50	0.57	0.49	0.46

EVA5	**MCC**	**MCC_H**	**MCC_E**	**MCC_C**
PHDpsi	0.61	0.69	0.62	0.54
PROFsec	0.65	0.72	0.66	0.58
SSPro2	0.67	0.73	0.67	0.60
YASPIN	0.65	0.72	0.66	0.59
JNET	0.62	0.68	0.62	0.57
PSIPRED	0.68	0.74	0.68	0.61

The overall Q3 and SOV prediction accuracies of PHDpsi, PROFsec, SSPro2, YASPIN, JNET and PSIPRED on the 188 sequences in the EVA5 common test set are listed in Table 5.1b and plotted in Figure 5.3 with their significant error margins. The Q3 prediction accuracy results for separate secondary structure elements (H, E and C) showed that PSIPRED and SSPro2 were the best in the prediction of helix with no significant differences between themselves, while YASPIN was significantly better than the remaining methods. In addition, YASPIN was significantly better at strand prediction than all other methods. The above observations were also confirmed by the SOV scores. However, the Matthew's correlation coefficients showed that YASPIN and PROFsec are equivalent in prediction quality (Table 5.2b) suggesting that the prediction error types made by each method are not accurately reflected in the Q3 and SOV scores.

Closer investigation of the types of errors made by each method on the EVA5 test set (Figure 5.4) showed that all methods are more or less missing out strand segments at the same rate (EU). On the other hand, PSIPRED and SSPro2 more

frequently over-predict, while the rest under-predict, helix segments (Hlo/Hlu). YASPIN's prediction scores mainly suffer from relatively frequently mistaking helices for strand (HW), over-elongating strand segments (Elo) and keeping helices too short (Hlu).

5.4.2 YASPIN position-specific reliability measures

The reliability-scoring scheme applied in YASPIN correlated well with the average secondary structure prediction accuracy (Q3). The relationship between the assigned reliability scores and their corresponding average prediction accuracy was almost linear. This means that the YASPIN confidence-scoring scheme accurately describes the reliability of each prediction. In approximately 48% of the predicted residues showing a confidence value of 5 or greater, 90% were accurately predicted (see Figure 5.2).

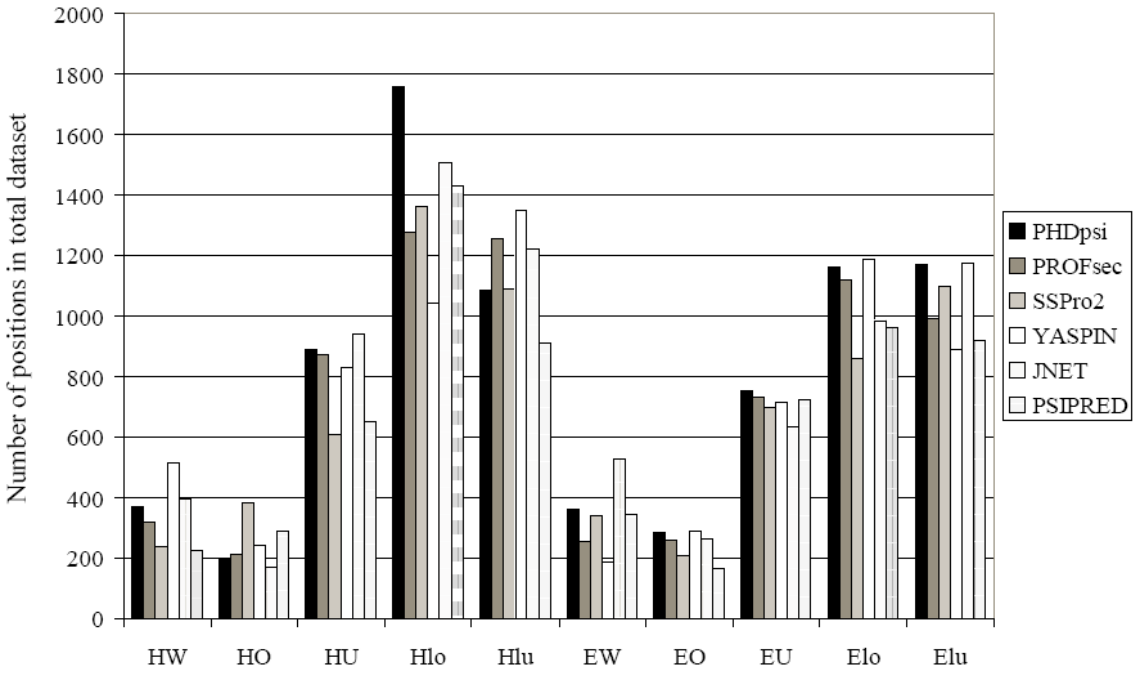


Figure 5.4. The extent of errors made by each of PHDpsi, PROFsec, SSPro2, YASPIN, JNET and PSIPRED on the independent EVA5 common dataset (188 sequences). (H/EW: wrong prediction (H→E, E→ H); H/EO: helix or strand structure over-predicted; H/EU: helix or strand structure under-prediction; H/Elo: helix or strand structure length errors due to over-prediction; and H/Elu: helix or strand structure length errors due to under-prediction).

5.4.3 The YASPIN Server

YASPIN is freely available online at the Bioinformatics Unit website (http://ibivu.cs.vu.nl/programs/yaspinwww/) at the Vrije University in Amsterdam and will also be mirrored at the Division of Mathematical Biology website (http://mathbio.nimr.mrc.ac.uk/) at the NIMR in London. The YASPIN server can perform predictions using a protein sequence or an already existing PSSM.

In addition, YASPIN has been integrated into the automated secondary structure prediction initiative of the EVA Server (Koh et al., 2003) for continual assessment of its prediction capabilities.

5.5. DISCUSSION

The difference between YASPIN and classical NN-based programs, such as JNET (JPRED), PSIPRED, SSPro2, PROFsec and PHDpsi, is its Hidden Neural Network (HNN) model (Krogh and Riis, 1999). It is worth noting that in the original HNN paper (Krogh and Riis, 1999), the NN and HMM components of the HNN model were trained in combination, while in later approaches including YASPIN, the NN and HMM have been trained separately. The latter training mode has also recently been applied to an HNN model for the prediction of protein residue contacts (Martelli et al., 2002).

In YASPIN, the initial predictions from the sequence-to-structure network are 7-state predictions of protein local structures, instead of the commonly used 3-state. The importance of this is that termini of secondary structure elements (SSEs), especially helices, have statistically significant different composition from other parts of the protein sequence (Richardson and Richardson, 1988; Serrano and Fersht, 1989). The network used in YASPIN is trained to capture these differences and provide the additional information via producing these 7-state predictions. Furthermore, the HMM that optimises these predictions before they are transformed to secondary structure is much simpler than the layers of networks in other programs. YASPIN is capable of modelling higher order relationships between SSEs, since it finds a global best solution for the segmentation of the sequence into SSEs. Its prediction accuracy is comparable to existing top performing methods and the program is much faster.

The classic approach of defining protein local structures as 3-state secondary

structures has recently been questioned (Pollastri et al., 2002; Karchin et al., 2003). One problem is that about 50 percent of residues are regarded as parts of random coil, except for some that are found in distinct local structures. In addition, the amino acid composition of alpha helices and strands varies enormously. Efforts have been made to obtain finer classifications of local structures. For example, the I-sites library defines some of these sequence-structure motifs by clustering sequence segments from a non-redundant database of known structures (Han and Baker, 1996; Bystroff and Baker, 1998). In this approach, a HMM (HMMSTR) was implemented to describe the transitions between these motifs (Bystroff et al., 2000). This Markov model was also used for the prediction of protein secondary structures. However, its performance was not as good as some of the NN-based programs. HMMSTR tried to capture the recurrent local features of both protein sequences and protein structures in a single model. Sequence information was mostly represented as the amino acid preferences at different sites of motifs, rather than being memorized in NNs. This model was much more complex than the Markov model employed in YASPIN, which records transition probabilities of local structures only.

YASPIN does not use an alignment algorithm directly, but uses the information as encoded in the PSSM that can be generated by PSI-BLAST (Altschul et al., 1997; Altschul and Koonin, 1998) or any other alignment program. Prediction of local structure is performed using a NN, like many NN-based programs. However, the targets of our NN prediction are 7-state local structures, rather than the common 3-state secondary structures targeted in most NN-based programs. This way, more structural information can be obtained via the sequence-to-structure network. A problem with our model (see Figure 5.1) is that strands predicted by YASPIN must be of at least 3 residues as well. According to the DSSP definition, β-bridges can often have only one residue. To overcome this problem, two different Markov models were designed, each having fewer states of strand structures than those currently used (Eb, E and Ee), but the prediction accuracy dropped (data not shown). This suggests that the sequence signals of the strand termini are important for the prediction.

The current YASPIN implementation is a predictor designed for the traditional 3-state secondary structure definitions. However, the architecture of the HNN model makes it very easy to adopt the program to predict local structures with different

classifications.

5.6. ACKNOWLEDGEMENTS

We would like to thank Dr. Jens Kleinjung for useful discussions and reading of the manuscript, Prof. Burkhard Rost for providing us with the PHD and PROFsec source codes, Dr. Gianluca Pollastri for the SSPro2 code, the JNET authors for making their method freely available online and the Vrije University Amsterdam for funding this project.

Chapter 6

Homology-extended sequence alignment

The content of this chapter is published in Simossis VA, Heringa J (2005) Simossis VA, Kleinjung J, Heringa J (2005) Homology-extended sequence alignment. Nucleic Acids Res. 33(3): 816-824.

6.1. ABSTRACT

We present a profile-profile multiple alignment strategy that uses database searching to collect homologues for each sequence in a given set, in order to enrich their available evolutionary information for the alignment. For each of the alignment sequences, the putative homologous sequences that score above a pre-defined threshold are incorporated into a position-specific pre-alignment profile. The enriched position-specific profile is used for standard progressive alignment, thereby more accurately describing the characteristic features of the given sequence set. We show that owing to the incorporation of the pre-alignment information into a standard progressive multiple alignment routine, the alignment quality between distant sequences increases significantly and outperforms state-of-the-art methods such as T-COFFEE (Notredame et al., 2000) and MUSCLE (Edgar, 2004). We also show that although entirely sequence-based, our novel strategy is better at aligning distant sequences if compared to a recent contact-based alignment method. Therefore, our pre-alignment profile strategy should be advantageous for applications that rely on high alignment accuracy such as local structure prediction, comparative modelling and threading.

6.2. INTRODUCTION

Protein sequences mutate to varying degrees of divergence through evolution. In order to identify homologous proteins and reveal important similarities, sequence alignment methods are commonly used (for recent overview see (Simossis et al., 2003)). These methods rely mainly on approximated evolutionary models that aim to reflect as accurately as possible the evolutionary paths that connect two or more protein sequences. Most state-of-the-art alignment methods align sequence pairs by dynamic programming (Needleman and Wunsch, 1970) and for three or more sequences they apply the progressive strategy (Feng and Doolittle, 1987), where sequences (or profiles) are hierarchically aligned in pairs according to a pre-generated tree (dendrogram), based on sequence similarity. However, when aligning the sequences or profiles to estimate their sequence similarity, pre-determined substitution scores are commonly employed (e.g. the scores from the BLOSUM (Henikoff and Henikoff, 1992) and PAM (Barker et al., 1978) series and more recently the JTT (Jones et al., 1992), GONNET

(Gonnet et al., 1992), VT (Muller and Vingron, 2000) and VTML (Muller et al., 2002) series) that have been derived using a specific set of "true" alignments. Such a generalisation presents a problem because these substitution scores reflect a standardised evolutionary model and introduce inconsistencies when applied to non-standard cases (Yu et al., 2003). As a result, although the similarity detection between closely related sequences is mostly unaffected by these inconsistencies and produces high-confidence alignments, sequences in the so-called "twilight zone" (<30% sequence identity) are extremely hard to align. This is because the evolutionary scenario relating them becomes virtually undetectable due to the noise introduced by the extent of mutational change that has occurred (Rost, 1999).

Improvements to the alignment of distant sequences have been achieved using several approaches. The evolutionary model describing the relation of a set of sequences can be re-adjusted to fit the sequence set and not an extrapolated generic model. Recently, Yu et al (2003) showed that the use of organism-specific or alignment set-specific background frequencies for contextual re-adjustment of the standard amino acid exchange weights provide a more sensitive and biologically accurate way to align sequences (Yu et al., 2003). Alternatively, structural or homologous sequence information can be incorporated into the alignment process to help identify the distant relations between sequences. The benefits of using related sequence information has been shown in numerous profile-profile alignment methods that apply different profile-scoring schemes (Jaroszewski et al., 2000; Rychlewski et al., 2000; Yona and Levitt, 2002; Ginalski et al., 2003; Mittelman et al., 2003; Sadreyev et al., 2003; von Ohsen et al., 2003; Capriotti et al., 2004; Chung and Yona, 2004; Edgar and Sjolander, 2004a; Ginalski et al., 2004; Soding, 2004; Tomii and Akiyama, 2004; von Ohsen et al., 2004; Wang and Dunbrack, 2004). Many of these scoring schemes have been assessed in recent comparison studies and have shown little significant difference in their respective performances (Edgar and Sjolander, 2004b; Ohlson et al., 2004). However, most of the profile-profile alignment approaches to date have been used mainly for sequence database searching (local pair-wise alignment). Multiple alignment methods that use profile information can be separated into two main groups: a) methods that are given a set of more than two sequences and return these sequences in aligned form; and b) methods that take a single sequence as input and collect related sequences by

aligning them to that sequence (profile-building). The DbClustal method (Thompson et al., 2000) belongs to the second group because it takes a single sequence as input and uses database-searching to collect homologous sequences for that single sequence. This newly built multiple alignment profile is then used to derive "anchor" points to guide the realignment of the query and homologous sequences using ClustalW (Thompson et al., 1994). Conversely, the profile pre-processing strategy of the PRALINE alignment method (Heringa, 1999) belongs to the first group, as it creates pre-alignment profiles for *each* sequence in a given set by adding information from all other sequences in the set. The method we present in this paper also belongs to the first group of multiple alignment methods. It takes two or more sequences as input, for each of which profiles are generated by database searching and then these profiles are used as starting input for progressive multiple alignment. This application of profile-profile alignment is to our knowledge yet unexplored. Other methods incorporate structural-based information because structure is more conserved than sequence (Chothia and Lesk, 1986) and therefore, it remains relatively unchanged through evolution, despite the mutational changes of the residues. Structural input has been used in the form of derived or predicted secondary structure (Chothia and Lesk, 1986; Heringa, 1999, 2000; Ginalski et al., 2003; Ginalski et al., 2004) and more recently in the form of side-chain contact information derived from tertiary protein structures, by using contact mutation probability matrices (Lin et al., 2003) in contact-based alignment (Kleinjung et al., 2004).

In this paper we present an application of profile-profile alignment for progressive multiple alignment, implemented in PRALINE$_{PSI}$. Pre-alignment profiles (pre-profiles) are generated using each sequence in a set as a PSI-BLAST (Altschul et al., 1997; Altschul and Koonin, 1998) query. The resulting PSI-BLAST local alignments are filtered for redundancy and converted to PRALINE pre-profiles, which replace the single sequence input that would otherwise be used for the alignment. For further details on the PRALINE alignment algorithm see (Heringa, 1999, 2002; Simossis and Heringa, 2003; Simossis et al., 2003). This extension of the pre-profile information beyond the sequences in the given set increases the information in the pre-profiles, and the new homologous sequences that are detected act as intermediary steps in the evolutionary paths that connect the sequences in the set. As a result, the increased

sensitivity of our method in detecting similarities becomes more evident, the more distant the sequence pairs become (or sequence-profile and profile-profile pairs in multiple sequence alignment).

6.3. MATERIALS AND METHODS

PRALINE$_{PSI}$ is written in the 'ANSI C' programming language. All programs were run on using locally installed versions of PSI-BLAST (Altschul et al., 1997; Altschul and Koonin, 1998), PRALINE (Heringa, 1999; Simossis and Heringa, 2003), ALICAO (Kleinjung et al., 2004), T-COFFEE v2.03 (Notredame et al., 2000) and MUSCLE v3.51 (Edgar, 2004).

6.3.1. The PRALINE$_{PSI}$ algorithm

Here we concentrate on the PRALINE$_{PSI}$-related features of the PRALINE multiple sequence alignment tool (Figure 6.1). Further details on PRALINE and what options it provides can be found in (Heringa, 1999, 2002; Simossis and Heringa, 2003; Simossis et al., 2003).

Generating PSI-BLAST pre-profiles. Each member of a sequence set is successively submitted as a query to a protein sequence database of choice, using PSI-BLAST. The iteration number and e-value cut-off threshold for PSI-BLAST can be manually set to any real number and are part of the quality-control of the hits that will be included in the pre-alignment profiles (pre-profiles). If the e-value threshold is too stringent and returns no hits or only redundant hits, PSI-BLAST is automatically restarted with a higher e-value tolerance in 10-fold increments (e.g. from 10^{-6} to 10^{-5} etc). Each resulting PSI-BLAST local alignment is filtered for redundant hits (100% sequence identity) and converted into a PRALINE pre-profile. The pre-profiles replace the single-sequence input of the basic PRALINE strategy (PRALINE$_{BASIC}$) (Heringa, 1999).

To test the sensitivity of PRALINE$_{PSI}$ to the content of the pre-profiles we run PSI-BLAST with fixed e-value thresholds 0, 10^{-6}, 10^{-3}, 10^{-2}, 1, 5 and 10. Note that for this test the automatic e-value threshold increments were switched off to allow

meaningful comparison between the results of each fixed threshold benchmark.

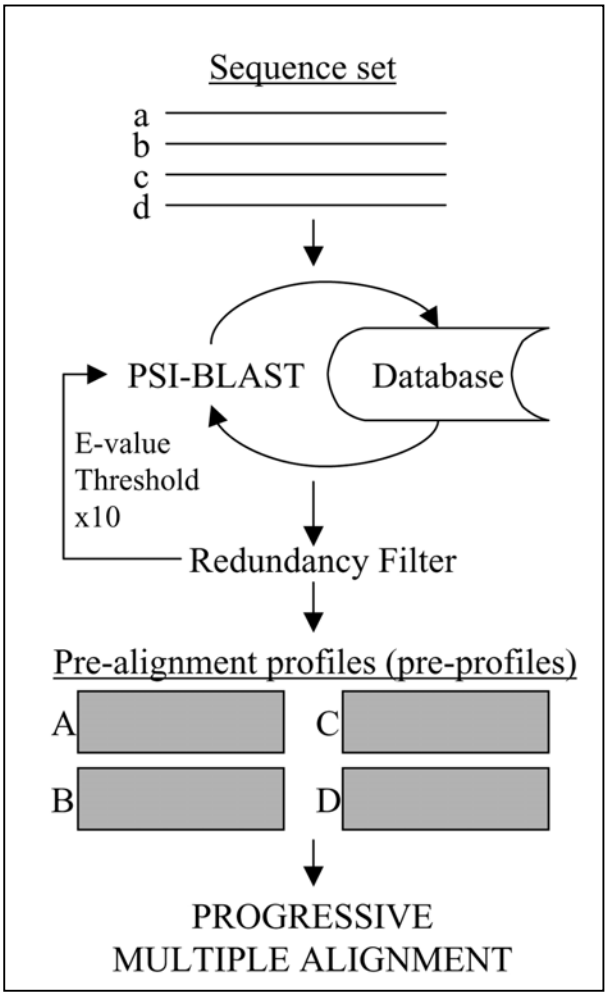


Figure 6.1. The schematic representation of the PRALINE$_{PSI}$ strategy. Each sequence is submitted as a PSI-BLAST query to a database of choice. The resulting local alignments are filtered for redundancy and if no hits are found or all hits are redundant, the search is re-run using a new e-value threshold 10 times less stringent. The final local alignments for each sequence are converted to a pre-profile and given to the PRALINE alignment algorithm.

Alignment hierarchy and tree construction. Similarly to the original PRALINE method, the alignment tree is not constructed prior to the progressive steps. First, all pre-profile pairs are scored using their alignment score and the closest two are aligned first. This new profile is then re-aligned to all the remaining pre-profiles and the next highest scoring pair is aligned, whether it is the new profile and a pre-profile or two separate pre-profiles. This continues until all sequences have been aligned and produces the final alignment tree.

Profile alignment. Since all sequence information is in profile form (pre-profiles or profiles), all dynamic programming alignment steps use the profile-profile scoring scheme. We define the score for a profile position (column) pair *x* and *y* as the sum of all residue pair scores adjusted according to the residue frequencies of that position:

$$Score(x,y) = \sum_{i}^{20}\sum_{j}^{20} \alpha_i \beta_j \log(\frac{p_{ij}}{p_i p_j}) \quad (1)$$

where α_i is the frequency with which residue *i* appears at position *x* and β_j is the frequency with which residue *j* appears at position *y*, p_{ij} is the frequency with which residues *i* and *j* appear aligned in the dataset used to derive the exchange weights matrix, p_i is the background frequency of residue *i* and p_j is the background frequency of residue *j*. Commonly, the *log()* component is simply the exchange weight provided by the selected log-odds substitution matrix (e.g. BLOSUM62).

6.3.2. Alignment method settings for benchmark

For the work described in this paper we searched a local version of the non-redundant database (NR) (August 2003 - 1,428,439 sequences) using PSI-BLAST with three iterations. The benchmarks were done using $PRALINE_{PSI}$ with a starting e-value threshold of 10^{-6}. The $PRALINE_{BASIC}$, profile pre-processing ($PRALINE_{PREPRO}$) and $PRALINE_{PSI}$ strategies of the PRALINE multiple alignment method were all run using the BLOSUM62 matrix and associated gap penalties 12 (gap-open) and 1 (gap-extension). For better comparison to $PRALINE_{PSI}$, the $PRALINE_{PREPRO}$ strategy was run so that all sequence set-related information was included in the pre-alignment profiles (pre-processing threshold 0 - not optimal). ALICAO (Kleinjung et al., 2004), T-COFFEE v2.03 (Notredame et al., 2000) and MUSCLE v3.51 (Edgar, 2004) were run using their default settings. The ALICAO method was only used in the HOMSTRAD (Mizuguchi et al., 1998) pair-wise alignment benchmark because it is not designed for multiple alignment.

The $PRALINE_{PSI}$ strategy has a high computational time compared to the other tested methods. This is due to the time PSI-BLAST needs to search over the non-redundant database, which on a current PC (IBIVU server Xeon 2.4GHz) averages to about 60 seconds per sequence.

6.3.3. Benchmark Datasets

HOMSTRAD. We separated the 1032 structure alignments in the HOMSTRAD dataset (Mizuguchi et al., 1998; Stebbings and Mizuguchi, 2004) (November 2003) into 633 pair-wise and 399 multiple alignment cases. We removed 9 of the pair-wise alignments to make the dataset comparable to the published ALICAO benchmark (Kleinjung et al., 2004). The final pair-wise set contained 624 alignments.

BAliBASE. We used reference sets 1-5 of BAliBASE 2.0 (Bahr et al., 2001) to explore the behaviour of $\text{PRALINE}_{\text{PSI}}$ in different alignment problem cases. Reference 1 is a set of 82 sequence sets that vary in relatedness and length but only contain relatively equidistant sequences. Reference 2 is a set of 23 alignment cases with one orphan sequence (outlier) amongst a group of related sequences. Reference 3 is a set of 12 alignment cases of two separate groups. References 4 and 5 hold 12 cases each, with N/C-terminal extensions and long internal insertions, respectively. The remaining reference sets 6, 7 and 8 were not used as they represent local alignment problem cases that the methods we are testing are not designed for.

6.3.4. Alignment quality assessment

The quality of the multiple alignments was assessed using both the sum-of-pairs (Q) and column (CS) scores, while the pair-wise alignments were assessed only using the sum-of-pairs (Q) score, taking the corresponding reference structure alignments as a standard of truth. For the Q score all correctly aligned residue pairs are expressed as a percentage of the total number of residue pairs in the alignment (no gapped positions).

$$Q = \frac{\text{Number of correctly aligned residue pairs}}{\text{Total number of aligned residue pairs in reference alignment}}$$

For the CS score all correct alignment positions (all residues of a whole alignment column) are expressed as a percentage of the alignment length.

$$CS = \frac{\text{Number of correctly aligned columns}}{\text{Total number of columns in reference alignment}}$$

The BAliBASE alignment cases were assessed using their core block annotations and the software provided by the BAliBASE authors. Some inconsistencies in the software calculations were corrected manually. For all other alignments we used

the VerAlign comparison software, which is available online (http://www.ibivu.cs.vu.nl/programs/veralignwww).

The sequence identities of the pair-wise and multiple alignments were calculated as the fraction of aligned identical residue pairs over the total number of aligned residue pairs in the reference structural alignments. The statistical significance of the Q and CS scores for the individual tested methods compared to PRALINEPSI was measured using the Kolmogorov-Smirnov Test that has been used in similar assessments (Notredame et al., 2000).

6.4. RESULTS

The benchmark assessment presented here has a two-fold objective. First, we compare the performance of multiple alignment methods in terms of their pair-wise and multiple alignment accuracy. Secondly, we test how the improvements of the pair-wise alignments transfer to that of the progressive multiple alignments.

6.3.5. Benchmark on pair-wise alignments

We used the 624 HOMSTRAD pair-wise alignments as a simple model to illustrate how the homology-extended information in the pre-alignment profiles (pre-profiles) affects similarity detection between sequences of different evolutionary distances.

For a meaningful assessment of $PRALINE_{PSI}$ performance using the incremental strategy from an e-value of 10^{-6} to a maximum of 10 (PSI-BLAST default setting), we set the alignment quality baseline to that of the basic dynamic programming strategy (sequence-sequence alignment) of PRALINE ($PRALINE_{BASIC}$) (Heringa, 1999) without profile pre-processing and only single-sequence input. To show the performance difference between the $PRALINE_{PSI}$ strategy and that of only using the sequences in a given set for enriching the information for dynamic programming, we aligned the sequence sets using the PRALINE profile pre-processing strategy (profile-profile alignment) ($PRALINE_{PREPRO}$) (Heringa, 1999).

We also compared the quality of the $PRALINE_{PSI}$ alignments to those produced by the contact-based method ALICAO (Kleinjung et al., 2004) and the latest versions

of the top-performing alignment methods T-COFFEE (Notredame et al., 2000) and MUSCLE (Edgar, 2004). It is important to clarify that these latter methods use the given sequence information only and are in a strict sense not fairly comparable to the profile-profile methods described above. However, although this is an unfair comparison, results from other MSA methods are essential for our study of how the pair-wise accuracy affects that of the progressive multiple alignment. Since there are no multiple alignment programs that use the profile-profile alignment strategy to compare to in the following sections (except $PRALINE_{PREPRO}$), we chose to compare $PRALINE_{PSI}$ against the best and increasingly popular multiple alignment methods available, namely T-COFFEE and MUSCLE. The comparison is reasonable and interesting since the $PRALINE_{PSI}$ strategy processes additional sequence information obtained via database searches in the background to help align a set of query sequences

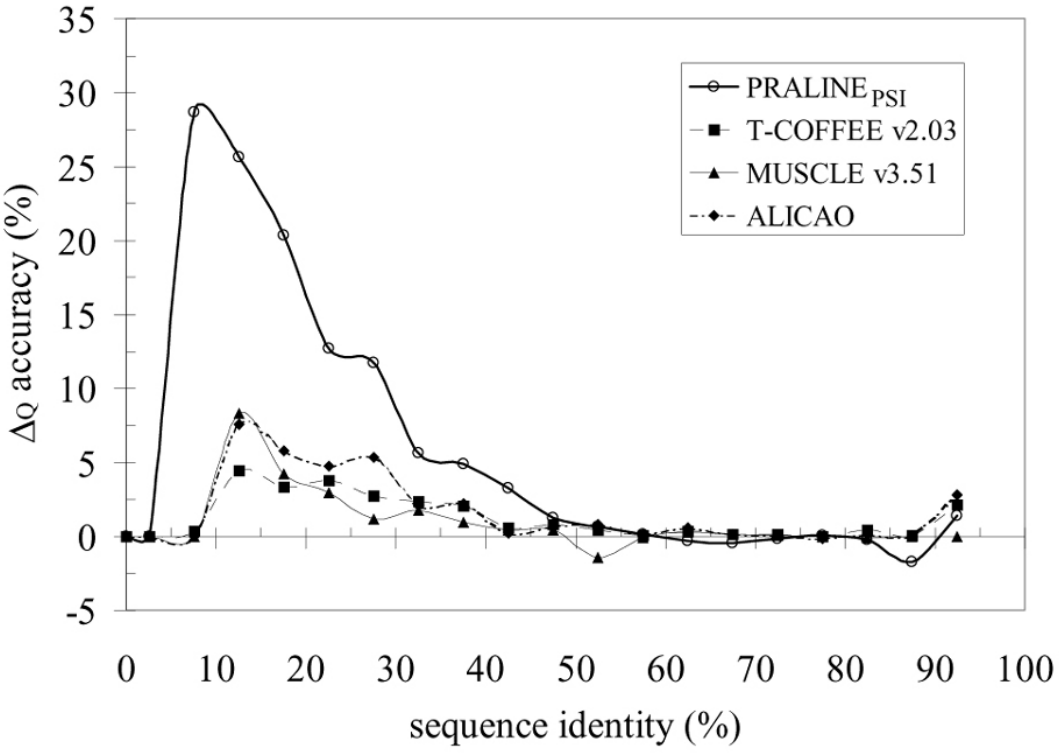


Figure 6.2. Comparison of alignment methods on the 624 HOMSTRAD pair-wise alignments (Q score). The difference (Δ) between the average scores of each tested alignment method and that of the $PRALINE_{BASIC}$ method is taken at 5%-intervals. The $PRALINE_{PREPRO}$ values for the pair-wise alignments are identical to those of $PRALINE_{BASIC}$ and therefore, they are not included. The $PRALINE_{PSI}$ scores are for the incremental strategy starting with an e-value of 10^{-6}.

in the foreground. This effectively means that $PRALINE_{PSI}$ takes a set of unaligned sequences as input and generates a multiple alignment of that same set as output, as do methods such as T-COFFEE and MUSCLE. In addition, the profile-profile pair-wise alignment methods that are currently available are all local alignment programs and therefore cannot be directly compared to our global approach. However, the log-average profile-scoring scheme (von Ohsen et al., 2003) can be applied to global alignment strategies and is used in MUSCLE as a log-expectation score (Edgar, 2004), where position-specific gap penalties are added to the original log-average scoring function.

Table 6.1. The sum-of-pairs (Q) scores of the 624 pair-wise alignment HOMSTRAD test cases. Scores are listed separately for sequence identity ranges of 0-30%, 30-60%, 60-100% and the overall scores with their standard deviation (numbers in brackets are the number of alignments each range contains). The "Δ Overall", "Improved" and "Worsened" columns are with reference to the baseline $PRALINE_{BASIC}$ scores and the last column "*P*" shows the statistical significance (*P* value from Kolmogorov-Smirnov Test) of the overall results of each method compared to those of $PRALINE_{PSI}$. *P*-values below 0.05 are underlined. The $PRALINE_{PREPRO}$ scores were not included because due to the lack of extra information (only 1 extra sequence per profile) they were identical to those of the $PRALINE_{BASIC}$ strategy.

METHOD	0-30 (227) %	30-60 (297) %	60-100 (110) %	All (624) %	Δ Overall (624) %	Improved %	Worsened %	*P*
$PRALINE_{BASIC}$	57.4	89.5	98.5	79.4 ± 23.2	-	-	-	$\underline{\leq 1e^{-4}}$
$PRALINE_{PSI}$	73.2	92.7	98.0	86.6 ± 16.6	7.1	65.2	15.5	-
$T\text{-}COFFEE_{v2.03}$	60.8	90.8	98.7	81.3 ± 22.0	1.8	50.5	23.2	$\underline{0.001}$
$MUSCLE_{v3.51}$	60.6	90.1	98.6	80.9 ± 21.8	1.4	43.9	22.9	$\underline{\leq 1e^{-4}}$
ALICAO	62.9	90.7	98.7	82.0 ± 21.6	2.6	51.0	18.4	$\underline{0.003}$

The difference (Δ) in Q scores compared to the $PRALINE_{BASIC}$ strategy is plotted as a function of sequence identity in Figure 6.2. Owing only to the incorporation of the homology-extended information in the pre-profiles, the difference in alignment quality (Δ_Q) is significantly higher compared to the $PRALINE_{BASIC}$ strategy. $PRALINE_{PSI}$ was also significantly better than the other tested methods and improved the most (>65%) and worsened the fewest (<14%) alignment cases, compared to the $PRALINE_{BASIC}$ method (Table 6.1). By far the largest improvement was observed in alignment cases with less than 30% identity (0-30%), although some cases between 30-60% were also significantly improved. As could be expected, the alignments above

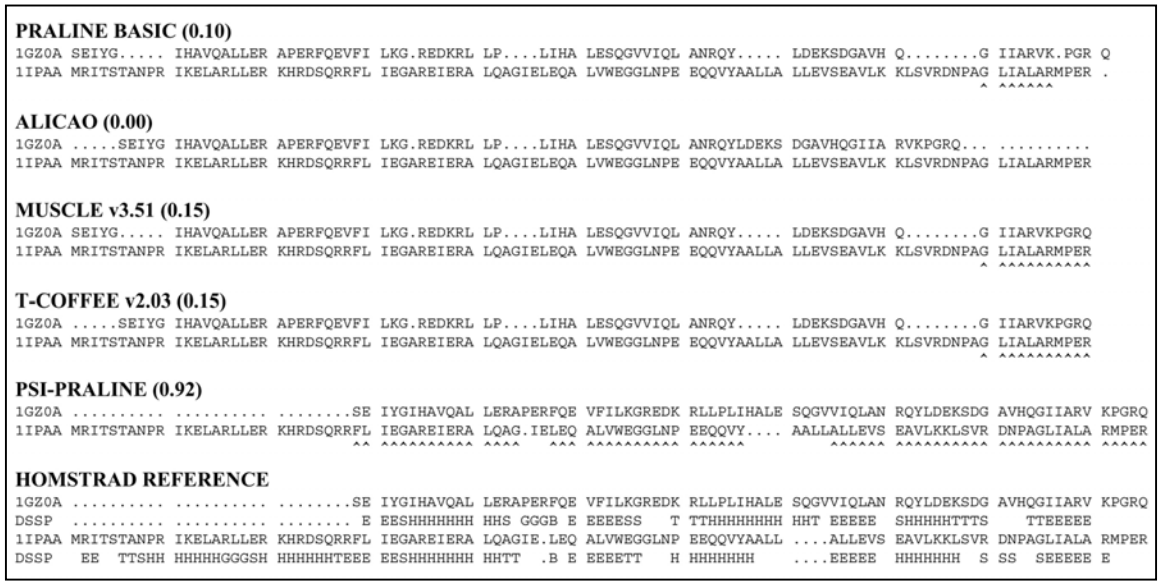


Figure 6.3. Sequence alignments of the protein methyltransferase (HOMSTRAD family "SopU_methylase_N"). The numbers in parentheses represent the Q scores of each alignment. The bottom alignment (HOMSTRAD) is the reference alignment derived from structure super-positioning and shows the secondary structures (DSSP-derived). Both the contact-based and the single sequence-based methods show a shift in the matched secondary structure elements, which is entirely prevented by the use of the extended evolutionary information. Correctly aligned residue pairs are denoted by a '^' sign.

60% sequence identity (60-100%) were relatively unaffected, albeit the overall quality slightly dropped, but not significantly (~0.5%).

An example of how the extended evolutionary information improves pair-wise alignment quality of distant sequences is illustrated in Figure 6.3. The methyltransferase enzyme alpha chains (HOMSTRAD family "SpoU_methylase_N") from *E.coli* (top sequence) and *T.thermophilus* share 16.7% sequence identity but have the same α/β knot fold. The very low similarity at the amino acid level causes a register-shift in the alignments of both the single-sequence and contact-based methods. This is entirely prevented by using the homology-extended information in the $\text{PRALINE}_{\text{PSI}}$ pre-profiles of each sequence and has allowed the correct alignment of the true related regions of these proteins. As a result, the $\text{PRALINE}_{\text{BASIC}}$ alignment is dramatically improved to over 90% accuracy. The small regions that have been misaligned do not affect the correct alignment of the structural elements of the fold, illustrated by the secondary structure elements of the sequences derived with DSSP (Kabsch and Sander, 1983), in the HOMSTRAD alignment.

6.3.6. Sensitivity to pre-profile information

We investigated how the stringency of homology-extension balances with the extent of improvement it can provide. PSI-BLAST was invoked using e-value thresholds of 0, 10^{-6}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 5 and 10 (PSI-BLAST default setting) to determine at which point the allowance of false positive hits in exchange for including more information became detrimental to $\text{PRALINE}_{\text{PSI}}$.

The use of the homology-extended pre-profiles has the same beneficial effect on similarity detection nearly irrespective of the e-value threshold used (Figure 6.4A). However, although the overall improvement is almost the same for all thresholds tested, the individual correlations of the Q scores of each threshold over the 624 cases show that there is some variation in the results (Table 6.2). In particular, e-value thresholds 10 and 5 seem to lead to lower alignment quality (Q scores) (Figure 6.4A). It is clear that the ease of admission of sequences (e-values from 0 to 10) can have an effect on individual cases, although with a minimum correlation coefficient of 0.83, the effect is not dramatic. It is possible that due to the strictness of the threshold, the method would fall back to $\text{PRALINE}_{\text{BASIC}}$ more often and as a result, correlate more with the 0 threshold results. However, the overall distribution of improved, unchanged and worsened alignment cases (Figure 6.4B), in combination with the relatively similar correlation of all thresholds to the $\text{PRALINE}_{\text{BASIC}}$ scores is very similar over the e-value thresholds taken. This suggests that a high stringency threshold is adequate to produce good quality alignments and in the cases where no hits or only redundant hits are returned, less stringent thresholds are stable enough to increment too.

Next, we re-activated the incrementing of the e-value threshold when no hits or only redundant hits were returned and assessed the quality of the alignments produced by $\text{PRALINE}_{\text{PSI}}$ with a starting e-value threshold of 10^{-6} to a maximum of 10 (Figure 6.4 - inc). It is important to note that the “inc” column has no occurrences of non-hit or only-redundant PSI-BLAST alignments. Therefore, the percentage of unaffected cases it contains serves as a baseline, further supporting that the distributions of the other thresholds are not greatly biased by the algorithm dropping back to $\text{PRALINE}_{\text{BASIC}}$. The incremental strategy covers all alignment cases and shows that the use of the homology-extended information in the pre-profiles greatly improves alignment quality, compared to the basic PRALINE method.

Table 6.2. The correlations between the Q scores of the 624 pair-wise alignments of HOMSTRAD aligned by $PRALINE_{PSI}$ using different e-value thresholds. The 0 threshold is equivalent to the $PRALINE_{BASIC}$ strategy.

Threshold	**10**	**5**	**1**	**10^{-1}**	**10^{-2}**	**10^{-3}**	**10^{-6}**	**0**
10	1.00							
5	1.00	1.00						
1	0.98	0.98	1.00					
10^{-1}	0.98	0.98	1.00	1.00				
10^{-2}	0.97	0.98	0.99	0.99	1.00			
10^{-3}	0.97	0.97	0.98	0.99	0.99	1.00		
10^{-6}	0.93	0.94	0.95	0.95	0.95	0.96	1.00	
0	0.83	0.83	0.84	0.84	0.84	0.84	0.85	1.00

Profile with most extra sequences from database

Profile with no extra sequences from database

It is understandable that since we have applied a common e-value threshold to all cases, the stringency will cause some sequences to lose useful input and others to incorporate false information. Ideally, one would run each alignment case with its optimum threshold. We investigated the theoretical upper performance limit of $PRALINE_{PSI}$, by executing each alignment case at its optimum threshold, except 0, and its potential benefits are shown in the "max" dataset results of Figure 6.4. Although this *a priori* selection is fictitious, the incremental strategy does not score very far below this upper limit.

6.3.7. Benchmark on multiple alignments

The progressive strategy for multiple alignment is in fact a hierarchical series of pair-wise alignments. Therefore, since the incorporation of external information in the form of pre-profiles allows better detection of relations between pairs of distant sequences, it should also produce more accurate multiple alignments.

We investigated the effects of using homology-extended information on the 399 HOMSTRAD multiple alignments. $PRALINE_{PSI}$ was run as described for the pair-wise alignments above. Alignment quality was assessed using both the Q and CS scores, the latter being the stricter of the two, using the HOMSTRAD structure alignments as a reference. All parameters were kept the same with the only difference being the information content of the pre-alignment profiles.

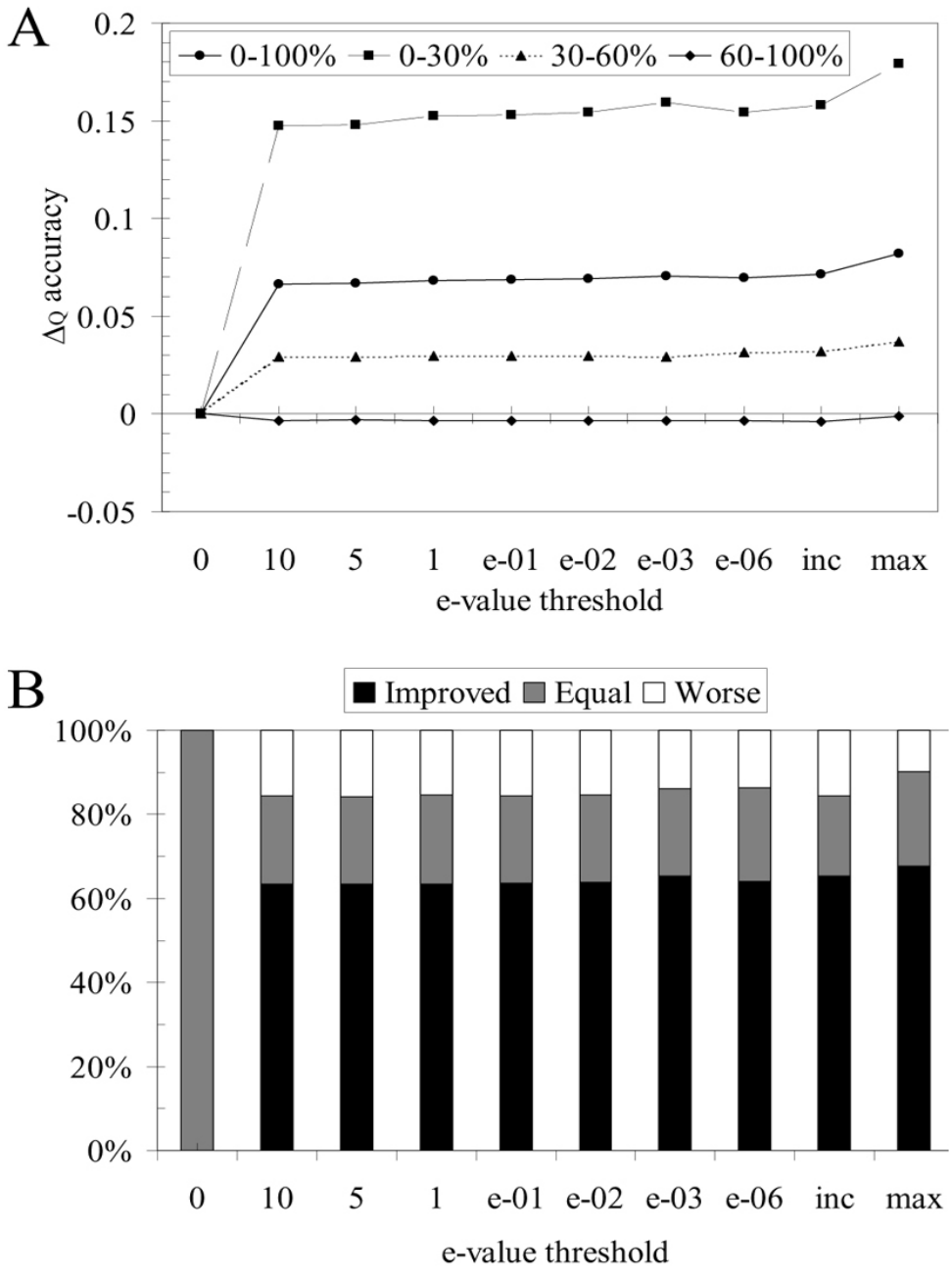


Figure 6.4. The effects of using e-value thresholds of increasing stringency in PRALINE$_{PSI}$ on the 624 HOMSTRAD pair-wise alignments. (a) The difference (Δ) between the average Q scores of PRALINE$_{PSI}$ and the basic PRALINE method, for all cases (0-100% sequence identity) and separately, cases between 0-30%, 30-60% and 60-100% sequence identity. (b) The distributions of improved, equal and worsened cases compared to the basic PRALINE method for each e-value threshold. The "inc" column is the PRALINE$_{PSI}$ incremental strategy starting from a threshold of 10^{-6} and the "max" column is PRALINE$_{PSI}$'s theoretical upper limit for the tested threshold range.

Table 6.3. The column (CS) and sum-of-pairs (Q) scores of the 399 mutiple alignment HOMSTRAD test cases. The scores are listed separately for sequence identity ranges of 0-30%, 30-60%, 60-100% and the overall scores with their standard deviation (numbers in brackets are the number of alignments each range contains). The "Δ Overall", "Improved" and "Worsened" columns are with reference to the baseline $PRALINE_{BASIC}$ scores and the last column "*P*" shows the statistical significance (*P* value from Kolmogorov-Smirnov Test) of the overall results of each method compared to those of $PRALINE_{PSI}$. *P*-values below 0.05 are underlined.

Column scores (CS)

METHOD	0-30 (121) %	30-60 (241) %	60-100 (37) %	All (399) %	Δ Overall (399) %	Improved %	Worsened %	*P*
$PRALINE_{BASIC}$	49.8	77.2	97.4	70.7 ± 22.1	-	-	-	$\underline{<1e^{-4}}$
$PRALINE_{PREPRO}$	50.2	77.6	97.5	71.1 ± 22.3	0.4	46.1	31.8	$\underline{<1e^{-4}}$
$PRALINE_{PSI}$	62.5	81.3	96.4	77.0 ± 19.6	6.3	70.2	17.0	-
$T\text{-}COFFEE_{v2.03}$	53.7	79.9	97.6	73.6 ± 20.9	2.9	62.2	25.6	$\underline{0.041}$
$MUSCLE_{v3.51}$	54.9	79.5	97.8	73.7 ± 20.8	3.0	62.4	23.1	$\underline{0.027}$

Sum-of-pairs scores (Q)

METHOD	0-30 (121) %	30-60 (241) %	60-100 (37) %	All (399) %	Δ Overall (399) %	Improved %	Worsened %	*P*
$PRALINE_{BASIC}$	60.4	85.4	98.4	79.0 ± 19.2	-	-	-	$\underline{<1e^{-4}}$
$PRALINE_{PREPRO}$	61.3	85.5	98.5	79.4 ± 19.6	0.3	49.1	31.6	$\underline{0.003}$
$PRALINE_{PSI}$	72.6	88.5	97.9	84.6 ± 15.7	5.5	72.4	16.3	-
$T\text{-}COFFEE_{v2.03}$	64.8	87.4	98.6	81.5 ± 17.8	2.5	63.7	27.3	$\underline{0.050}$
$MUSCLE_{v3.51}$	65.8	87.0	98.7	81.7 ± 17.4	2.6	65.2	21.8	$\underline{0.034}$

Consistent with the pair-wise results, when comparing the quality of the alignments produced by $PRALINE_{PSI}$ to that of the other multiple alignment methods, we observed a similar level of improvement (Figure 6.5). $PRALINE_{PSI}$ has the highest ratio of improved cases over worsened compared to the $PRALINE_{BASIC}$ strategy. Also, the overall alignment quality is either better than or comparable to the best of the other tested methods throughout all levels of sequence identity (Table 6.3). This is very interesting because although the T-COFFEE and MUSCLE alignment strategies are different to PRALINE and produce better alignments compared to the $PRALINE_{BASIC}$ strategy, they base their alignment only on the given sequence set-specific information. Conversely, $PRALINE_{PSI}$ is exactly the same algorithm as $PRALINE_{BASIC}$, the only

difference being the use of the homology-extended information.

Similarly to the pair-wise benchmark, $PRALINE_{PSI}$ produces better multiple alignments than all other tested methods, especially in the very distant cases. This shows that our initial assumption that the high level of pair-wise alignment quality would have a positive effect on multiple alignment was valid. Understandably, the level of improvement in alignment quality is not the same as in the pair-wise cases because multiple sequences share more complex inter-relations and the homology-extended information is not always ideal for the sequence or profile pairs. Since the optimisation strategies of T-COFFEE and MUSCLE can make very good use of sequence set-specific information, these methods would largely benefit as well if they would extend likewise the information they use.

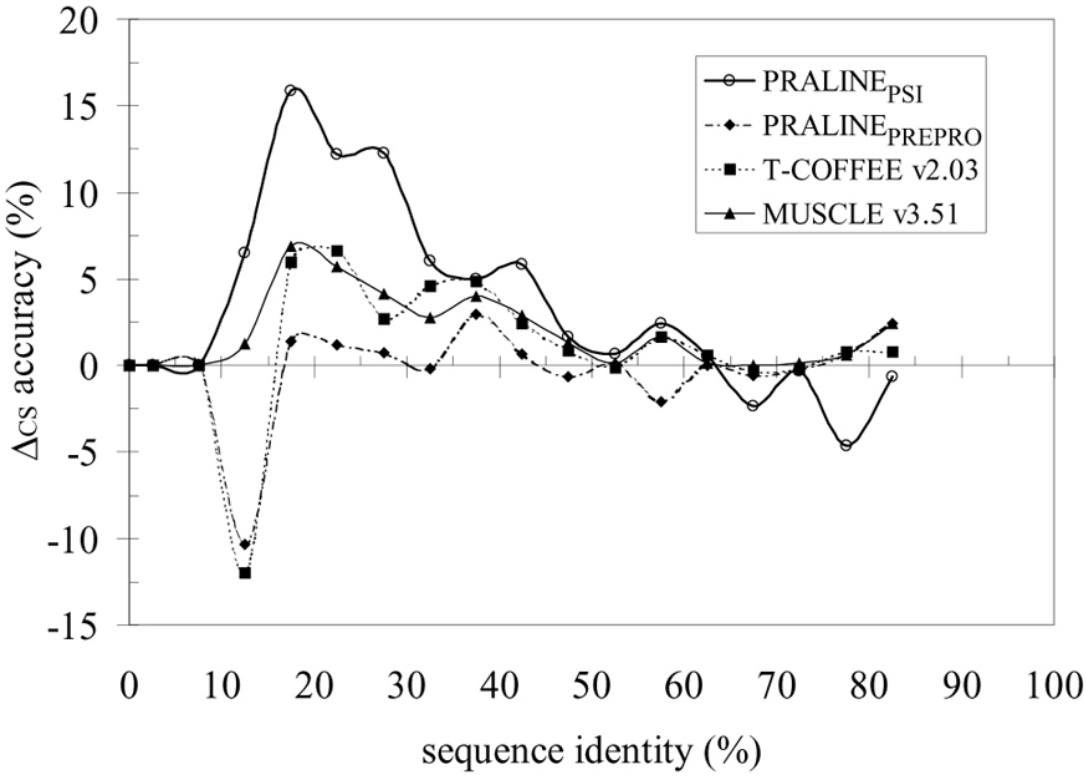


Figure 6.5. Comparison of alignment methods on the 399 HOMSTRAD multiple alignments (CS score). The difference (Δ) between the average scores of each tested alignment method and that of the $PRALINE_{BASIC}$ method is taken at 5%-intervals. The $PRALINE_{PSI}$ scores are for the incremental strategy starting with an e-value of 10^{-6}.

Table 6.4. The column (CS) and sum-of-pair (Q) scores of the BAliBASE test cases in references 1-5. The scores are listed separately for each reference set and the overall average, weighted relative to the number of alignments in each reference set (numbers in brackets are the number of alignments each set contains). The "*P*" columns show the statistical significance (*P* value from Kolmogorov-Smirnov Test) of the results of each method compared to $PRALINE_{PSI}$. *P*-values below 0.05 are underlined.

Column scores (CS)

METHOD	REF 1 (82) %	*P*	REF 2 (23) %	*P*	REF 3 (12) %	*P*	REF 4 (12) %	*P*	REF 5 (12) %	*P*	Weighted Average %	*P*
$PRALINE_{BASIC}$	76.9	0.425	51.0	0.593	54.0	0.786	38.5	0.991	59.8	0.786	66.0	0.187
$PRALINE_{PREPRO}$	78.4	0.425	56.2	0.842	50.8	0.786	30.7	0.991	77.1	0.786	68.3	0.949
$PRALINE_{PSI}$	83.9	-	61.0	-	55.8	-	53.9	-	68.6	-	73.9	-
$T\text{-}COFFEE_{v2.03}$	78.9	0.548	58.5	0.593	54.8	0.786	70.8	0.186	86.1	0.186	73.4	0.768
$MUSCLE_{v3.51}$	79.9	0.914	60.2	0.842	58.3	0.786	63.3	0.186	91.4	0.066	74.4	0.858

Sum-of-pairs scores (Q)

METHOD	REF 1 (82) %	*P*	REF 2 (23) %	*P*	REF 3 (12) %	*P*	REF 4 (12) %	*P*	REF 5 (12) %	*P*	Weighted Average %	*P*
$PRALINE_{BASIC}$	85.0	0.319	91.0	<u>0.017</u>	77.1	0.991	73.2	0.991	82.5	0.786	84.1	<u>0.030</u>
$PRALINE_{PREPRO}$	86.0	0.425	93.1	0.593	77.9	0.991	74.1	0.991	88.9	0.991	85.7	0.858
$PRALINE_{PSI}$	90.4	-	94.0	-	76.4	-	79.9	-	81.8	-	88.2	-
$T\text{-}COFFEE_{v2.03}$	86.2	0.425	93.9	0.842	76.7	0.786	88.3	0.433	94.6	0.186	87.5	0.858
$MUSCLE_{v3.51}$	87.0	0.914	93.7	0.842	79.6	0.433	88.9	0.186	97.8	<u>0.019</u>	88.5	0.928

6.3.8. Behaviour to specific alignment problems

The HOMSTRAD alignment sets enable us to test the effects of the homology-extended information on alignments of varying difficulty, but the averaged sequence identity values for the multiple alignments did not discern between specific alignment problems biologists and bioinformaticians are faced with, i.e. two sequence sets with low average sequence identity could be for example, a closely related group plus one orphan or two distant groups of closely related sequences. Therefore, we used the BAliBASE multiple alignment benchmark set to test how $PRALINE_{PSI}$ performs on specific alignment cases of known composition. Similarly to the HOMSTRAD benchmark, the BAliBASE sets were aligned with and without homology-extended information and the $PRALINE_{PSI}$ alignments were also compared to results from T-COFFEE and MUSCLE that are to date the highest scoring methods on the BAliBASE reference alignment sets.

It is important to note that BAliBASE is critically small and as the *P* values

from the Kolmogorov-Smirnov Test show, the statistical significance of most of the results on BAliBASE presented here are too low to allow confident conclusions to be drawn (Table 6.4).

Overall, the alignment cases of reference 1, 2 and 3 comprise over 80% of the alignment cases in BAliBASE and contain most of the distantly related sequences (based on average sequence identity). Our results show that the use of the homology-extended information in these distant sequence cases (>100 alignments) consistently improves the alignment quality compared to the basic PRALINE method, albeit the improvement is not as high as that of T-COFFEE and MUSCLE in the 24 alignment cases in references 4 and 5 (Table 6.4). Considering the alignment cases of the two latter sets (long insertions and terminal extensions), the differences in the improvement levels are mainly results of the distinct gap weighting of the individual alignment methods. Nonetheless, such alignment cases can be easily detected by the difference in sequence lengths and therefore, a user would be encouraged to use the MUSCLE or T-COFFEE methods when aligning such sequence sets.

6.5. DISCUSSION

The use of profiles to store evolutionary information improves alignment quality and has been known for some time now. One of the most famous examples has been the transition of BLAST to the more accurate PSI-BLAST database-searching tool and more recently to numerous database search tools that use profile-profile alignment strategies. However, although this highly successful technique allowed the correct detection of very distant homologues, it is not included in top-performing multiple alignment methods. In this paper we have shown that the dramatic benefits of using homology-extended information for pair-wise alignment is stably sustained through the progressive steps of multiple alignment. This suggests that there is information to be extracted from residue sequences before extending to structure, for which the available data remains limiting.

The $\text{PRALINE}_{\text{PSI}}$ strategy can positively affect the field of database searching, which is one of the most important computational areas in biological research. With $\text{PRALINE}_{\text{PSI}}$ we are able to detect similarities between distant sequences with a higher accuracy, but we also use database searching as our means of collecting the extended

information. In iterative alignment-based search tools such as QUEST (Taylor, 1998; Taylor and Brown, 1999), this introduces an optimisation scenario that allows the use of the search hits for better alignment before they are used for the next step.

The PRALINE$_{PSI}$ strategy does not intervene with further alignment optimisations such as the re-adjustment of amino acid substitution matrices (Yu et al., 2003), profile-profile scoring techniques (Rychlewski et al., 2000; Yona and Levitt, 2002; Pei et al., 2003; Sadreyev et al., 2003; von Ohsen et al., 2003; Capriotti et al., 2004; Edgar and Sjolander, 2004a; Soding, 2004; Wang and Dunbrack, 2004) and the incorporation of contact or structural information (Ginalski et al., 2003; Ginalski et al., 2004). Since the extended information is in the form of a profile, contact and structural information can be readily incorporated to further enrich the position specific information for the alignment. Furthermore, the alignment routine still uses substitution matrices and therefore the re-adjustment strategies are applicable. Finally, all pair-wise alignments in both pair-wise and multiple alignment cases are in the profile-profile form, allowing for any profile-scoring technique to be applied. Therefore, homology-extended sequence alignment should be used together with the aforementioned alignment optimisations in current and future multiple alignment methods.

As would be expected, PRALINE$_{PSI}$'s use of the PSI-BLAST search engine over a database as large as the non-redundant (NR) makes its computational time much higher than that of fast methods such as MUSCLE. However, since the development of software such as IMPALA (Schaffer et al., 1999), a sequence can be used to search a position-specific profile database rather than the much larger sequence databases, making the inclusion of appropriate profiles much faster and less CPU intensive. Also, the large size of the pre-profiles that sometimes contain over 1000 sequences creates a bottleneck at the progressive all-against-all alignment steps. Nonetheless, since the PRALINE code has been parallelised (Kleinjung et al., 2002), the PRALINE$_{PSI}$ strategy computational time can be improved.

More importantly, for fields that rely on very high alignment accuracy such as comparative modelling, secondary structure prediction, threading and detection of evolutionary relationships, the improvement in alignment accuracy is far more important than the speed at which the alignments are generated. A significantly better alignment of two or more distant sequences can provide answers to questions that do

not rely on speedy solutions. Considering the apparent success of using profile-profile alignment beyond the pair-wise stage, we expect that more multiple sequence alignment algorithms will employ homology-extended profile information instead of single sequence input as starting points for the progressive strategy.

6.6. AVAILABILITY

The PRALINE$_{PSI}$ strategy is part of the freely available PRALINE WWW Server at http://ibivu.cs.vu.nl/programs/pralinewww. The PRALINE source code can be made available upon request.

6.7. ACKNOWLEDGEMENTS

We would like to thank the Vrije Universiteit for funding this project. Thanks are also due to the authors of the software and databases we have used for making them freely available online and two anonymous referees for their constructive comments. Funding to pay the Open Access publication charges of this article was provided by the Vrije Universiteit Amsterdam.

Chapter 7

Improvement and Limitations of Secondary Structure-Guided Multiple Alignment Quality

---

The content of this chapter is pending publication as Simossis VA, Heringa J (2005) Improvement and limitations of secondary structure-guided multiple alignment quality. Bioinformatics (submitted).

7.1. ABSTRACT

Many current homology detection strategies use profiles and predicted secondary structure information to improve alignment quality of distantly related sequences, but the improvement level is often limited by prediction quality. We tested the effects of integrating secondary structure information into the scoring scheme of a standard global multiple alignment method and a new profile-profile global multiple alignment strategy. The integration of secondary structure information within the context of multiple alignment significantly improves the overall alignment quality related to the standard multiple alignment method and that of distant sequences relative to the profile-profile multiple alignment technique. We also show that the limitation in the level of improvement compared to using "true" secondary structure information is the result of specific types of prediction errors that most state-of-the-art prediction methods consistently make.

7.2. INTRODUCTION

In recent years, the detection of homologies between distant sequences has been significantly improved through profile-profile local alignment (Jaroszewski *et al.*, 2000; Rychlewski *et al.*, 2000; Yona and Levitt, 2002; Ginalski *et al.*, 2003; Mittelman *et al.*, 2003; Sadreyev and Grishin, 2003; Sadreyev *et al.*, 2003; von Ohsen *et al.*, 2003; Capriotti *et al.*, 2004; Edgar and Sjolander, 2004a; Ginalski *et al.*, 2004; Soding, 2004; Tomii and Akiyama, 2004; von Ohsen *et al.*, 2004; Wang and Dunbrack, 2004). In these approaches, single sequence input is enriched with homologous position-specific information. This enriched information can be represented either as a profile or a hidden Markov model (HMM) and two profiles or HMMs or a combination of the two can be aligned using different profile-profile scoring schemes. Recent comparison studies of such scoring schemes (Edgar and Sjolander, 2004b; Ohlson *et al.*, 2004) suggest that the scoring scheme based on information theory used in *prof_sim* (Yona and Levitt, 2002) and COMPASS (Sadreyev and Grishin, 2003) are the most sensitive.

Many of these profile-profile alignment methods have recently incorporated structural information into their profile-profile scoring schemes to further increase the detection of homologies (Ginalski *et al.*, 2003; Chung and Yona, 2004; Ginalski *et al.*,

2004; Soding, 2004; von Ohsen *et al.*, 2004). The reason for the reported success of this incorporation is that the level of evolutionary conservation of structure is higher than that of sequence, making it more robust to evolutionary changes (Chothia and Lesk, 1986), such that the structural information can successfully anchor the alignment of distantly related sequences. The most reliable secondary structure information comes from the three-dimensional co-ordinates of solved crystal structures using the defined hydrogen-bonding patterns (Kabsch and Sander, 1983; Frishman and Argos, 1995). However, since solved structure information is limited, prediction methods are commonly invoked, the most popular being PSIPRED (Jones, 1999) [for recent secondary structure prediction review see (Simossis and Heringa, 2004b)]. As a result, the quality of the predicted secondary structure becomes a limiting factor for the level of homology detection improvement, compared to that reached using the "true" structure information (Chung and Yona, 2004).

In recent work, we have shown that the advantages of pair-wise profile-profile alignment is directly transferable to progressive multiple alignment strategies (Simossis *et al.*, 2005). The use of profiles for each of the query sequences that are enriched with sequence information generated from database searching at the start of a progressive procedure produces higher quality multiple alignments than only using the sequences in the given set. In this paper we introduce, as an addition to this strategy, a profile-scoring scheme for multiple sequence alignment that integrates secondary structure information using the Lüthy secondary structure-specific substitution matrices (Lüthy *et al.*, 1991), as originally proposed by Heringa in 2000 (Heringa, 2000). We find that this integration of predicted secondary structure-specific parameters into the profile-scoring scheme of a standard progressive multiple alignment strategy (Heringa, 1999) and the recent homology-extended multiple alignment strategy (Simossis *et al.*, 2005) improves alignment quality between distant sequences (<30% sequence identity). However, consistent with the results from the pair-wise local alignment studies previously mentioned, the improvement level is limited compared to that possible when using the "true" secondary structure. Therefore, we also investigate the reason for this limitation and find that despite the high per-residue (Q3) and segment overlap [SOV; (Zemla *et al.*, 1999)] accuracy scores of state-of-the-art prediction methods, the

limitation is a result of specific types of prediction errors these methods consistently make.

7.3. MATERIALS AND METHODS

All methods were run locally on the IBIVU server (Dual Intel Pentium Xeon 2.4GHz). The PRALINE tool is written in the C programming language. The strategies of the PRALINE alignment tool that have been tested in this study can all be freely used on the PRALINE WWW Server at http://www.ibivu.cs.vu.nl/programs/pralinewww/ and the source code can be made available upon request from the authors. Please note that for local installations, the secondary structure prediction methods and PSI-BLAST (Altschul *et al.*, 1997) must be obtained separately.

7.3.1. Algorithm

The PRALINE progressive alignment algorithm used for all strategies tested in this study is described in detail in previously published work (Heringa, 1999, 2000, 2002; Simossis and Heringa, 2003; Simossis *et al.*, 2005). All alignments that were not guided by secondary structure information were run using BLOSUM62 and associated gap penalties 12 and 1.

The addition to the PRALINE tool we present here is a profile-scoring scheme that integrates secondary structure-specific information in the form of secondary structure-specific substitution scores from the Lüthy series of matrices (Lüthy *et al.*, 1991). The scoring scheme has been integrated so that all available PRALINE strategies can optionally make use of secondary structure information.

In all the PRALINE strategies except PRALINE$_{PSI}$ (Simossis *et al.*, 2005) that uses extended-homology information, the secondary structure information for each sequence in a given set is associated with its corresponding sequence. In the homology-extended strategy of PRALINE$_{PSI}$, a PSI-BLAST (Altschul *et al.*, 1997) search is invoked for each sequence in the given set to collect potential homologues that score above a predetermined e-value cut-off. This extended collection of homologues for each query sequence is represented as a homology-extended profile and used as the starting point for the progressive routine instead of only the individual query sequences

in the set. These profiles are also used as input for predicting the secondary structure for each of the query sequences. Then, the secondary structure information is assigned to all the hits in the profile generated from database searching. This way, each homologue in the homology-extended profile is assigned the secondary structure of the query (top) sequence. This generalisation of the local structure of the homology-extended profile sequences is necessary because re-running predictions for all of them would be computationally prohibitive and biologically uncertain given that homologous fragments detected by PSI-BLAST can be relatively short. To do this we would have to re-run PSI-BLAST searches for each hit and do a prediction for each one. Considering that some homology-extended profiles contain up to 500 hits, this would increase computational time dramatically.

7.3.2. Profile Scoring

Normally, the PRALINE profile-scoring scheme uses the following equation to score a pair of profile columns *x* and *y*:

$$Score(x, y) = \sum_{i}^{20} \sum_{j}^{20} \alpha_i \beta_j \log(\frac{p_{ij}}{p_i p_j}) \quad (1)$$

where α_i and β_j are the frequencies with which residues *i* and *j* appear in columns *x* and *y*, respectively, p_{ij} is the frequency with which residues *i* and *j* appear aligned in the dataset used to derive the exchange weights matrix, p_i is the background frequency of residue *i* and p_j is the background frequency of residue *j*. Commonly, the *log()* component is simply the exchange weight provided by the selected log-odds substitution matrix (e.g. the PRALINE default is BLOSUM62).

In the scoring scheme that integrates secondary structure information, the score for the matching of two profile columns becomes a combination of sequence- and structure-derived substitution score elements, as explained in the following section.

Essentially, if both amino acids being compared from each profile column belong to the same type of secondary structure element H, E or C, then the corresponding secondary structure-specific Lüthy substitution matrix (Lüthy *et al.*, 1991) is used. Otherwise, the BLOSUM62 matrix (or any other assigned matrix) is used. Given that the sum of all residue frequencies assigned a secondary structure class

(H, E or C) in each profile column adds to 1, the score *S* for any two profile columns *x* and *y* is given by the sum of all pair-wise scores in the following equation:

$$S(x,y)=\sum_{SSx=1}^{3}\sum_{i=1}^{20}\sum_{SSy=1}^{3}\sum_{j=1}^{20} f_{SSxi}\, f_{SSyj}\,\alpha_i\,\beta_j\left(\delta(SSx,SSy)Lüthy(i,j)+(1-\delta(SSx,SSy))M(i,j)\right) \quad (2)$$

where $\delta(SSx,SSy)=\begin{cases}1, SSx=SSy\\ 0, otherwise\end{cases}$; *SSx* and *SSy* denote the helix, strand or coil (H, E, C) assignments in profile columns *x* and *y*, respectively; f_{SSxi} and f_{SSyj} are the frequencies of amino acids *i* and *j* in profile columns *x* and *y*, respectively, that belong to a specific secondary structure class (H, E or C); α_i and β_j are the frequencies with which residues *i* and *j* appear in columns *x* and *y*, respectively; $Lüthy_{ss}(i, j)$ and *M(i, j)* are the Lüthy and sequence-based (e.g. the PRALINE default is BLOSUM62) substitution score for amino acid pair *i* and *j*, respectively.

The profile-scoring function works so that when alignment blocks are being compared, the usage of the BLOSUM62 and Lüthy matrices follows the mixtures of secondary structures observed in each block; for example, if the positions of two profiles *A* and *B* are being compared and *A* has observed secondary structures ‘H’ and ‘E’, while *B* has ‘C’ and ‘H’, then the corresponding usage of BLOSUM62 and Lüthy would be 75% (C-E, C-H, H-E) and 25% (H-H), respectively.

7.3.3. Weighting of exchange matrices

The residue exchange matrices were normalized by multiplication of all the values by a factor such that the occurrence-weighted diagonal elements (identities) added up to 1.0. We used the normalization scheme and residue frequencies described in (Abagyan and Batalov, 1997) (A 7.85, C 2.55, D 5.17, E 6.95, F 4., G 6.52, H 2.12, I 5.45, K 5.66, L 8.86, M 2.51, N 4.59, P 4.67, Q 4.09, R 5.17, S 7.1, T 5.48, V 6.2, W 1.46, Y 3.05). The weighting factors for the helix class (H), strand class (E) and coil class (C) Lüthy matrices were calculated as 0.27 (gap penalties 15.0 and 1.5), 0.26 (gap penalties 12.0 and 1.0) and 0.26 (gap penalties 12.0 and 1.0), respectively. For the BLOSUM62 matrix the weighting factor was calculated to be 0.89 (gap penalties 12.0 and 1.0).

7.3.4. Databases

We used the HOMSTRAD database (Mizuguchi *et al.*, 1998) of structure alignments (2004 update) as our reference. Due to the structure super-positioning of the sequences in the alignments, some sequences are missing those sections that could not be aligned in the structure. To avoid complications from erroneous secondary structure predictions due to these missing sections, from the 399 multiple alignments in the database, we selected only those alignments that contained full corresponding sequences to those in the PDB three-dimensional co-ordinate files (Berman *et al.*, 2000). The final dataset contained 254 multiple alignments of evenly distributed percentage sequence identities.

7.3.5. Secondary structure sources

The predicted secondary structures for the sequences in our datasets were obtained by the SSPro 2.01 (Pollastri *et al.*, 2002), PROFsec (Rost, personal communication), YASPIN (Lin *et al.*, 2005) and PSIPRED (Jones, 1999) prediction programs. All methods use PSI-BLAST profile information for their predictions, so we used the profiles generated by the incremental strategy of $PRALINE_{PSI}$ (Simossis *et al.*, 2005) as input for the predictions. In all cases the PSI-BLAST program was invoked using 3 iterations and a starting e-value cut-off of 10^{-6} on the non-redundant database (NR update 11/2004). In addition, since all the HOMSTRAD sequences have solved three-dimensional structure co-ordinates in the PDB, we used the DSSP software (Kabsch and Sander, 1983) to derive the "true" secondary structures of the dataset sequences. These "true" secondary structures were used to assess the quality of the predicted secondary structures (see below in Secondary structure prediction accuracy) and also to determine how the "true" local structure information affects alignment quality when used instead of the predictions.

7.3.6. Recording prediction error types

The prediction error types of PSIPRED, YASPIN, SSPro and PROFsec were recorded based on the scheme used in the benchmarking study of secondary structure prediction methods for fold recognition by McGuffin and Jones (McGuffin and Jones, 2003) for accurate comparison with their results: a) wrong prediction (w), b) over-

prediction (o), c) under-prediction (u) and d) length (l) errors. The length (l) errors were also recorded separately as over and under-predictions for comparison between the methods. The four error types are illustrated below for clarity.

```
AA      MDYFTLFGLPARYQLDTQALSLRFQQLAAVQTINQ…
SS              HHHH    EEEEE HHH HHHHH     …
DSSP      HHH   HHHH HHHHHHHHHH              …
Errors    uuu         uuuwwwwwl ll ooooo    …
```

7.3.7. Introducing errors into the DSSP secondary structure information

We used the DSSP-assigned secondary structure information for each sequence in the 254 HOMSTRAD sets to test how randomly induced prediction errors and systematic prediction error types affect alignment quality. To this end, we first randomised the DSSP information to different extents. Secondly, we induced the maximum level of each of three prediction error types into the DSSP information: wrong (E→H or H→E), over- (C→H or C→E) and under-predictions (H→C or E→C). These two approaches are described in more detail below.

(1) *Randomisation of DSSP information:* We first randomised the "true" secondary structure information from the DSSP assignments so that we generated secondary structure information with 0% (original assignment) to 100% error. For example, for 20% randomisation we randomly altered the assigned states at 20% of the sequence positions. We used drawing without replacement such that in a sequence of 100 residues 20 random positions were randomly changed. The state changes were each time randomly switched from the original state (e.g. H) to one of the remaining two states (e.g. E or C).

(2) *Simulating prediction error types*: To simulate wrong predictions we switched the helix- and strand-specific Lüthy matrices. This way, all amino acid pairs assigned helix (H) in DSSP are considered as strand (E) by the alignment method and *vice versa*. Similarly, for under-predictions of helix and strand we switched the helix- and strand-specific Lüthy matrices with that of coil, respectively. As a result, all amino acid pairs with helix or strand assignments are considered as coil by the alignment method. Finally, to simulate helix and strand over-predictions, the coil-specific Lüthy matrix was switched with the helix- and strand-specific matrices, respectively, so that

all amino acid pairs assigned coil in DSSP were accordingly considered either as helix or strand. It is important to note that these simulations retain the BLOSUM62 usage for amino acid pairs that do not share the same secondary structure state and therefore the resulting effects on the relative alignment quality are a direct result of the respective prediction errors.

In each case, the error-induced DSSP assignments were used to align the 254 HOMSTRAD sequences sets described in the Databases section. These new alignments were then compared to those generated by using the original predictions and the original DSSP assignments. The alignment quality and prediction accuracy assessment measures we used are described next.

7.3.8. Multiple alignment accuracy

The performance of the alignment methods tested were based on the sum-of-pairs (Q) and column score (CS) accuracy measures, using the HOMSTRAD structure alignments as a standard of truth. The calculations used to derive these scores are represented in the equations below:

$$Q = \frac{\text{Number of correctly aligned residue pairs}}{\text{Total number of aligned residue pairs in reference alignment}}$$

$$CS = \frac{\text{Number of correctly aligned columns}}{\text{Total number of columns in reference alignment}}$$

The alignments generated by integrating secondary structure information were compared to those generated by the corresponding strategies without secondary structure information as a delta (Δ) accuracy score. Positive values denote an improvement when using secondary structure information.

7.3.9. Secondary structure prediction accuracy

The accuracy of the predictions on the data were measured using the Q3 and SOV score measures (Zemla *et al.*, 1999), using the DSSP-derived secondary structures as a reference. To enable comparison with the 3-class alphabets of the prediction methods, the DSSP 8-state secondary structure representation (H, G, E, B, I, S, T, -) was grouped according to the 3-state scheme proposed by (Heringa and Argos, 1991;

Rost and Sander, 1993), i.e. H and G were considered as helix (H), E and B as strand (E), and all others as coil (C).

7.4. RESULTS

We integrated the use of secondary structure information into the scoring function of the multiple alignment tool PRALINE (Heringa, 1999) as described in the Algorithm section. We investigated the effects that the "true" and predicted secondary structure information have on the quality of multiple sequence alignments, using a standard alignment strategy (sequence-sequence) ($PRALINE_{BASIC}$) and the recent homology-extended multiple alignment strategy that builds a profile for each sequence through database-searching before starting the progressive scheme (profile-profile) ($PRALINE_{PSI}$) (Simossis *et al.*, 2005). The recent $PRALINE_{PSI}$ strategy represents a significant improvement over alternative methods tested. To distinguish the $PRALINE_{BASIC}$ and $PRALINE_{PSI}$ strategies that integrate secondary structure information, we refer to them as $PRALINE_{BASIC\text{-}SS}$ and $PRALINE_{PSI\text{-}SS}$, respectively.

7.4.1. Alignment benchmarks

We used 254 sequence sets from the HOMSTRAD reference alignment dataset and predicted the secondary structure of all 1162 sequences using the PROFsec, SSPro, YASPIN and PSIPRED prediction methods. Each sequence set was aligned using the $PRALINE_{BASIC\text{-}SS}$ and $PRALINE_{PSI\text{-}SS}$ strategies, so that each strategy generated one secondary structure-guided alignment using predicted secondary structure and one using DSSP-derived secondary structure. The alignments were compared to those produced by the corresponding strategies without the secondary structure information, in terms of their column (CS) and sum-of-pairs (Q) scores. Figure 6.1 illustrates the delta (Δ) CS scores of the $PRALINE_{BASIC\text{-}SS}$ and $PRALINE_{PSI\text{-}SS}$ strategies that used YASPIN, PSIPRED, SSPRO predictions and DSSP, compared to the $PRALINE_{BASIC}$ and $PRALINE_{PSI}$ strategies, respectively. PROFsec failed to complete all predictions and therefore the results of its integration were not included in the Figure 7.1 plots because they are not directly comparable to the alignment quality results.

The integration of predicted secondary structure information improved the $PRALINE_{BASIC}$ method by over 4% in the alignment of distant sequences with less than

60% sequence identity (Table 7.1). For the $PRALINE_{PSI}$ method, the use of secondary structure information in general was only significantly beneficial in very distant sequences (<20% sequence identity) (clearly apparent in Figure 7.1B), suggesting that the use of the homology-extended profiles already covers many of the limitations of standard alignment. The observed difference in achievable alignment improvement between predicted and 'true' secondary structure information was overall not significant. This is mainly due to the very high per-residue (Q3) prediction accuracy achieved by the prediction methods on this data (~81%), since all other parameters of the alignment method were kept identical for all benchmarks.

Table 7.1. Alignment quality of PRALINE methods using secondary structure information compared to the standard PRALINE method ($PRALINE_{BASIC}$) and the homology-extended alignment strategy ($PRALINE_{PSI}$). The "*P*" columns show the statistical significance (*P* value from Kolmogorov-Smirnov Test) of the results of each method on all alignment cases (0-100% sequence identity) compared to $PRALINE_{BASIC}$.

	0-100 (%)	**0-30 (%)**	**30-60 (%)**	**60-100 (%)**	***P* (0-100)**
Column score (CS)					
$PRALINE_{BASIC}$	63.8	38.7	68.5	95.5	-
$PRALINE_{BASIC\ -DSSP}$	67.7	44.6	72.1	95.8	0.196
$PRALINE_{BASIC\ -SSPRO}$	67.6	44.9	71.9	96.0	0.106
$PRALINE_{BASIC-YASPIN}$	68.0	45.3	72.2	96.3	0.106
$PRALINE_{BASIC-PSIPRED}$	67.4	43.5	72.1	95.9	0.337
$PRALINE_{PSI}$	70.2	50.2	73.6	96.7	0.025
$PRALINE_{PSI-DSSP}$	70.5	50.5	74.0	96.8	0.008
$PRALINE_{PSI\ -SSPRO}$	70.1	49.8	73.7	96.7	0.042
$PRALINE_{PSI-YASPIN}$	70.0	49.7	73.6	96.5	0.042
$PRALINE_{PSI-PSIPRED}$	70.1	50.2	73.5	96.7	0.014
T-COFFEEv2.01	67.6	44.0	72.2	95.8	0.237
MUSCLEv3.51	67.5	45.0	71.6	96.3	0.461
Sum-of-pairs (Q)					
$PRALINE_{BASIC}$	81.8	63.9	86.0	98.2	-
$PRALINE_{BASIC\ -DSSP}$	84.7	69.6	88.3	98.5	0.085
$PRALINE_{BASIC\ -SSPRO}$	84.6	70.1	88.0	98.6	0.101
$PRALINE_{BASIC-YASPIN}$	84.7	70.1	88.2	98.7	0.196
$PRALINE_{BASIC-PSIPRED}$	84.2	68.4	87.9	98.6	0.131
$PRALINE_{PSI}$	86.4	74.1	89.2	98.7	0.001
$PRALINE_{PSI-DSSP}$	86.5	74.1	89.4	98.7	0.002
$PRALINE_{PSI\ -SSPRO}$	86.5	74.3	89.2	98.7	0.001
$PRALINE_{PSI-YASPIN}$	86.4	73.9	89.2	98.6	0.002
$PRALINE_{PSI-PSIPRED}$	86.4	74.2	89.1	98.7	0.002
T-COFFEEv2.01	84.1	68.5	87.8	98.5	0.161
MUSCLEv3.51	84.2	69.9	87.4	98.6	0.237

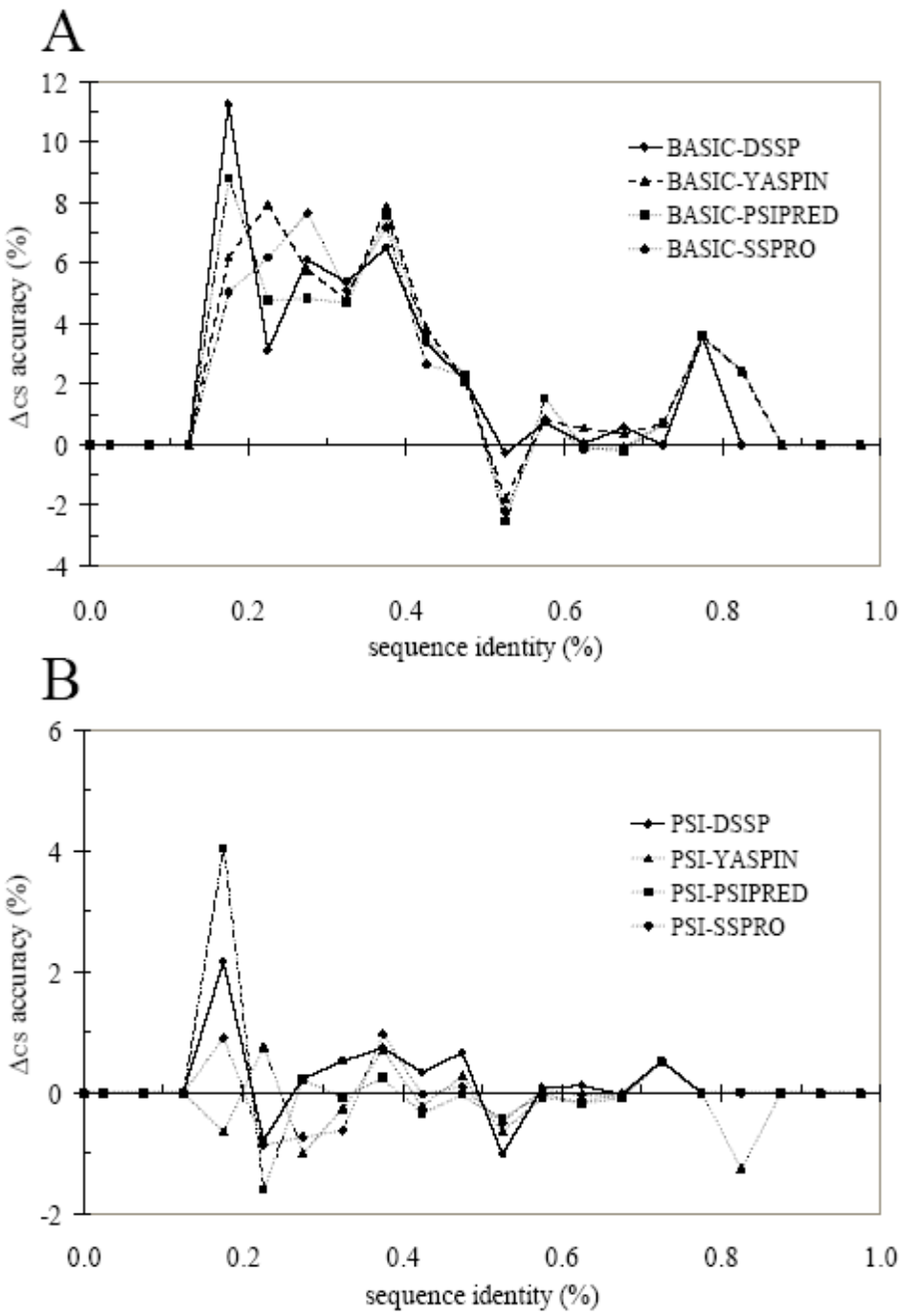


Figure. 7.1. The delta (Δ) column scores (CS) of the alignments produced by each strategy compared to those of A) $PRALINE_{BASIC}$ and B) $PRALINE_{PSI}$ on the 254 HOMSTRAD multiple alignments. The data points represent average CS scores for alignments in 5% sequence identity bins. (BASIC-DSSP, BASIC-YASPIN, BASIC-SSPRO and BASIC-PSIPRED: $PRALINE_{BASIC\text{-}SS}$ using DSSP, YASPIN, SSPRO and PSIPRED secondary structure information, respectively; PSI-DSSP, BASIC-YASPIN, BASIC-SSPRO and PSI-PSIPRED: $PRALINE_{PSI\text{-}SS}$ using DSSP, YASPIN, SSPRO and PSIPRED secondary structure information, respectively).

However, for both the PRALINE$_{BASIC-SS}$ and PRALINE$_{PSI-SS}$ alignment strategies in cases under 30% sequence identity, the differences in prediction quality clearly have affected the level of alignment quality improvement (Figure 7.1). Interestingly, the PRALINE$_{BASIC-SS}$ strategy shows higher improvement of the alignments between 20-40% sequence identity using predicted information rather than the "true" secondary structure (Figure1A). Similarly, the cases below 20% sequence identity aligned using PRALINE$_{PSI-SS}$ with PSIPRED predictions also improved more than when the "true" information was used (Figure 7.1B). It is important to make the distinction that although secondary structure information quality is judged by its accuracy, the way in which it affects the alignment of multiple sequences is a different matter. In particular, in very distant cases such as those below 30% sequence identity, although conserved at a high level, it is not always the case that structure is entirely conserved. Therefore, despite the fact that predicted secondary structures contain errors it may be that in some cases these errors result in a more consistent segmentation of the information, thus resulting in a better alignment of these distant cases. The factors that dictate 'good quality' predicted information for multiple alignment were further investigated in terms of specific prediction error types.

7.4.2. Prediction error analysis

The SSPro, PROFsec, YASPIN and PSIPRED predictions were scanned for error types as described in (McGuffin and Jones, 2003) and each error type was plotted as a percentage of the total number of residues in the data set (Figure 7.2A). In addition, we separated the data that corresponded to the <30% sequence identity region of the plots in Figure 7.1 and found an almost identical distribution of errors, albeit YASPIN's helix correct predictions were only second to PSIPRED's (Figure 7.2B). On the data we have used, the level and types of prediction errors made by either of these methods is very similar. We also tested a 'majority voting' and dynamic programming (DP) consensus approach (Simossis and Heringa, 2004a, 2005d), but the correlation between the methods' predictions was too high to allow any significant reduction in overall prediction errors (data not shown). These results suggest that although the

predictions are very similar, the prediction methods make consistently the same types of errors thus limiting the potential alignment quality improvement.

In order to measure the effects of different levels of prediction errors on alignment quality we randomised the DSSP assignments to different extents (0-100%) for all 1162 sequences and used that information to re-align the 254 sets (Figure 7.3).

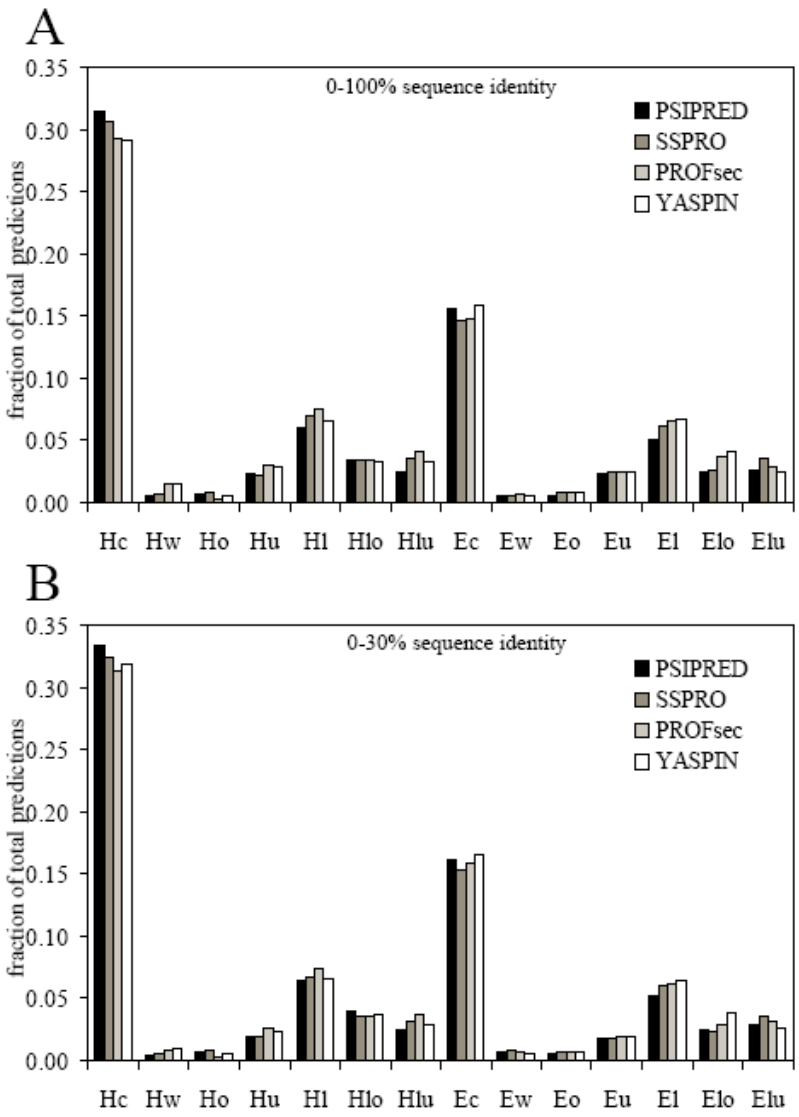


Figure. 7.2. Assessment of the predictions of PSIPRED, SSPRO, PROFsec and YASPIN on A) the 254 HOMSTRAD sequence sets (1162 sequences) and B) all sets with <30% sequence identity. The number of residues assigned to each class is expressed as a percentage of all recorded residues. (Hc: Helic correct; Hw: Helix wrong; Ho: Helix over-predictions; Hu: Helix under-predictions; Hl: Helix length errors; Hlo and Hlu: Helix length error subsets for over- and under-predictions, respectively). The same classification applies to strand (E).

As would be expected, the more we randomised the DSSP information (see Methods section), the more alignment quality dropped accordingly for both $PRALINE_{BASIC-SS}$ and $PRALINE_{PSI-SS}$. The main conclusion from Figure 7.3 is that the scoring scheme we have developed for secondary structure incorporation into a multiple alignment method clearly makes good use of the additional information, because randomised information is nothing but detrimental to it. In particular, this can be clearly seen by the difference in alignment quality improvement achieved by PSIPRED and DSSP20. Given that the PSIPRED predictions have ~81% per-residue accuracy they can be considered as equivalent to the DSSP20 information, which is also 80% correct, albeit the latter is known to contain random information. For both $PRALINE_{BASIC-SS}$ (Figure 7.3A) and $PRALINE_{PSI-SS}$ (Figure 7.3B), the use of the PSIPRED information gives clearly better alignments than DSSP20. Therefore, this suggests that the prediction errors that limit alignment improvement are systematic (non-random). In addition, the evident higher improvement achieved by $PRALINE_{PSI-SS}$ (Figure 7.3B) in the cases with less than 20% sequence identity using the PSIPRED information compared to the "true" secondary structure (DSSP0) and the equivalent DSSP20 underlines the point we raised in the previous section about predicted information being more consistently segmented, albeit not always accurate. We have shown here that the errors in predicted information are not random and that they are mostly edge over- and under-predictions. Next we assess how multiple alignment is affected by these types of errors.

In Figure 7.2 the most common prediction error types are length errors (Hl and El), with over- (Hlo and Elo) and under-predictions (Hlu and Elu) occurring at mostly similar frequencies. We investigated what would happen to the alignment quality if the DSSP assignments were treated so that all possible wrong, over- and under-prediction errors were induced (Figures 7.4 and 7.5). To this end, we performed 6 tests where we treated all DSSP helix assignments as strand (H-Wrong), all strands as helices (E-Wrong), all coil positions as helices (H-Over), all coil positions as strands (E-Over), all helix positions as coil (H-Under) and finally all strand positions as coil (E-Under). These 'induced' errors apply only to those residue pairs with matching secondary structure assignments and therefore, the results in Figures 4 and 5 represent what would happen to the alignment quality if all the matching structural positions were wrongly, over- or under-predicted as helix or strand, while all non-matched secondary structure

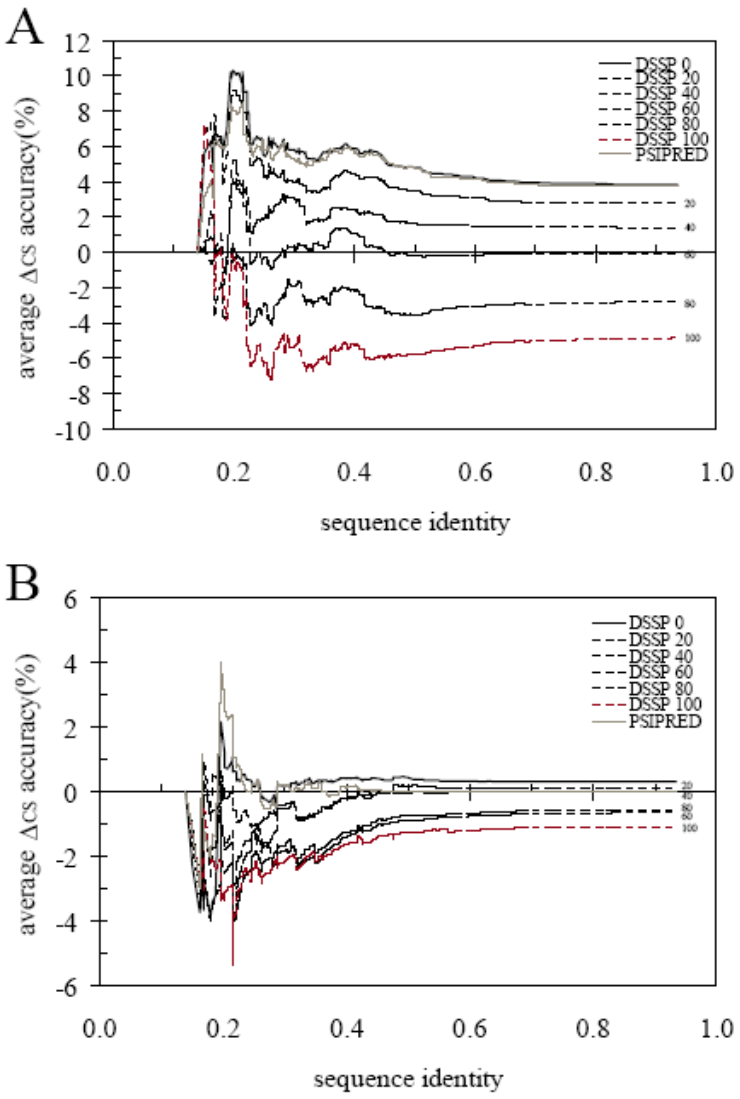


Figure. 7.3. The cumulative average delta (Δ) column (CS) score of the multiple alignments produced by **(A)** $PRALINE_{BASIC-SS}$ compared to those produced by the $PRALINE_{BASIC}$ and **(B)** $PRALINE_{PSI-SS}$ compared to those produced by the $PRALINE_{PSI}$, as a function of sequence identity. $PRALINE_{BASIC-SS}$ and $PRALINE_{PSI-SS}$ are given DSSP secondary structure information randomised to different degrees (DSSP20 indicates 20% of the residue assignments are randomised) to guide the alignment. Each data point represents the average improvement of all data preceding it, i.e. the last data point represents the overall improvement.

elements were still scored with BLOSUM62. Due to the very similar results between the prediction methods we have only used the PSIPRED results to compare against. In the case of the $PRALINE_{BASIC-SS}$ strategy, the induction of systematic errors limits the possible alignment quality improvement level but is interestingly still better than the standard strategy (Figures 7.4 and 7.5). Conversely, for $PRALINE_{PSI-SS}$ the over-prediction of secondary structure elements is not just limiting, but becomes detrimental

for the alignment quality (Figures 4 and 5). This makes sense because in the $PRALINE_{PSI\text{-}SS}$ strategy the majority of the homology-extended information is concentrated in helical or strand regions and therefore, by treating the intervening and much less enriched coil regions as part of the same element, due to the over-predictions, induces segment shift errors in the alignment.

Based on these results, we can conclude that the limitation of improvement is not a random event and does not suffer greatly by wrong prediction errors (E→H or H→E). The main source of the limitation is the reduced use of the Lüthy matrices due to over- (C→H or C→E) and under-predictions (H→C or E→C) that increase structural class mismatches between matched residue pairs. Also, in the case of the homology-extended strategy, elongation of secondary structure elements causes regional shifts and generates worse alignments than without using the predicted information.

7.5. DISCUSSION

The use of homology-extended profiles in progressive multiple alignment has already been shown to significantly outperform currently available state-of-the-art methods in distant cases (Simossis *et al.*, 2005). In this paper we have shown that the use of 'true' and predicted secondary structure information can improve both standard and homology-extended multiple alignment accuracy even further. However, the level of improvement that the structure-specific scoring scheme presented here can achieve is directly dependent on the segmentation quality of the predicted information and suffers from systematic prediction errors that are made by all current state-of-the-art prediction methods. In particular, the main source of improvement limitation was the large number of structurally non-matching residue pairs, so increasing the consistency of the per-position predictions could reduce this limitation. We have attempted to do this by generating a consensus prediction for each sequence, but due to the high correlation levels between the predictions from the four tested methods, no significant improvement was possible.

Interestingly, the same prediction error types have also been shown to be the reason for limited performance in fold recognition methods that integrate predicted secondary structure information (McGuffin and Jones, 2003). Therefore, it is a valid

suggestion that the secondary structure prediction field should focus on reducing these errors in the future.

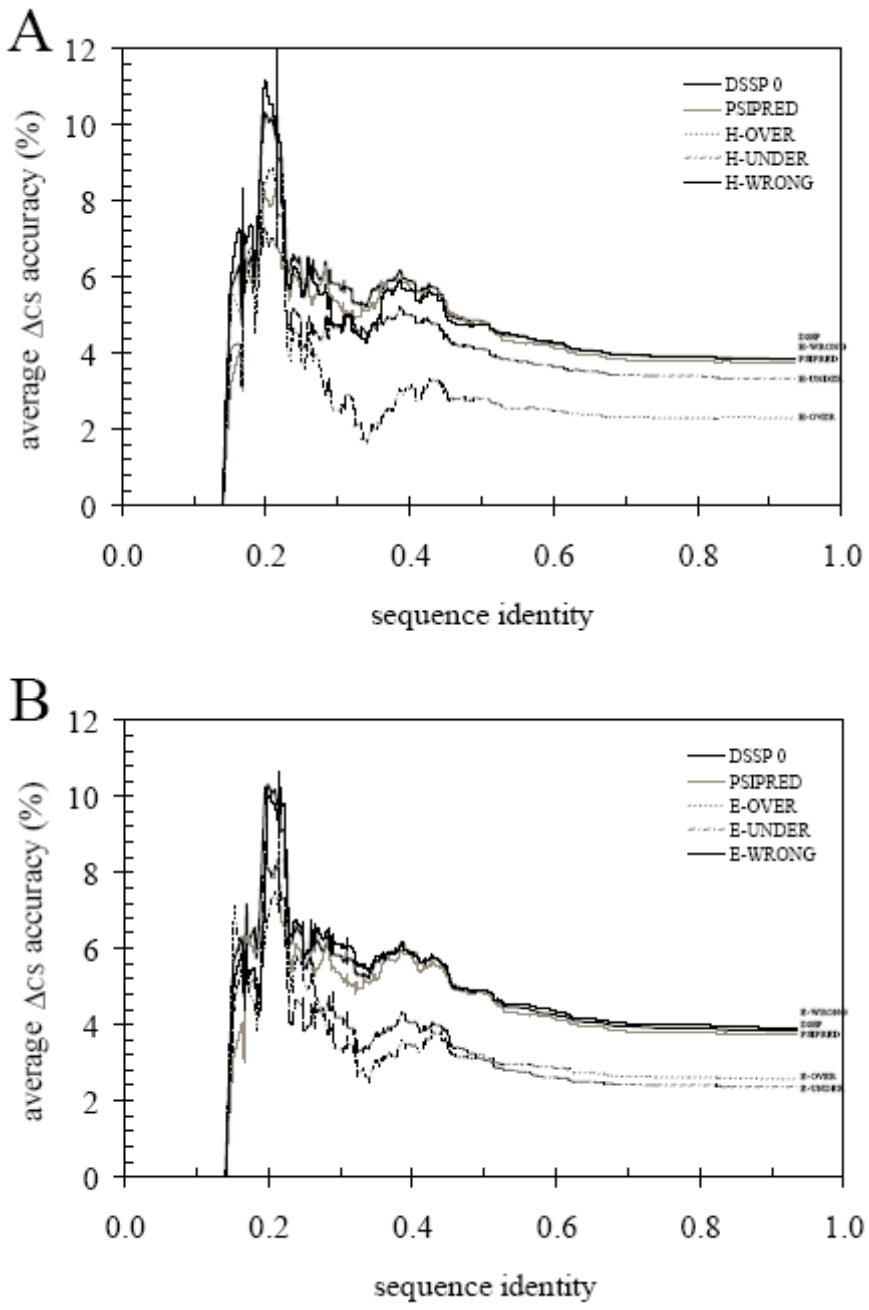


Figure. 7.4. The cumulative average delta (Δ) column (CS) score of the multiple alignments produced by $PRALINE_{BASIC\text{-}SS}$ compared to those produced by the $PRALINE_{BASIC}$ as a function of sequence identity. $PRALINE_{BASIC\text{-}SS}$ is given DSSP (DSSP) and PSIPRED (PSIPRED) secondary structure information to guide the alignments. In addition, the DSSP assignments are modified with (A) helix-specific and (B) strand-specific errors. H-OVER and E-OVER correspond to alignments using DSSP secondary structure information where helix and strand over-predictions are maximized, respectively. H-UNDER and E-UNDER correspond to alignments using DSSP secondary structure information where helix and strand under-predictions are maximized, respectively. H-WRONG and E-WRONG correspond to alignments using DSSP secondary structure information where helix and strand switches are maximized, respectively. Each data point represents the average improvement of all data preceding it, i.e. the last data point represents the overall improvement.

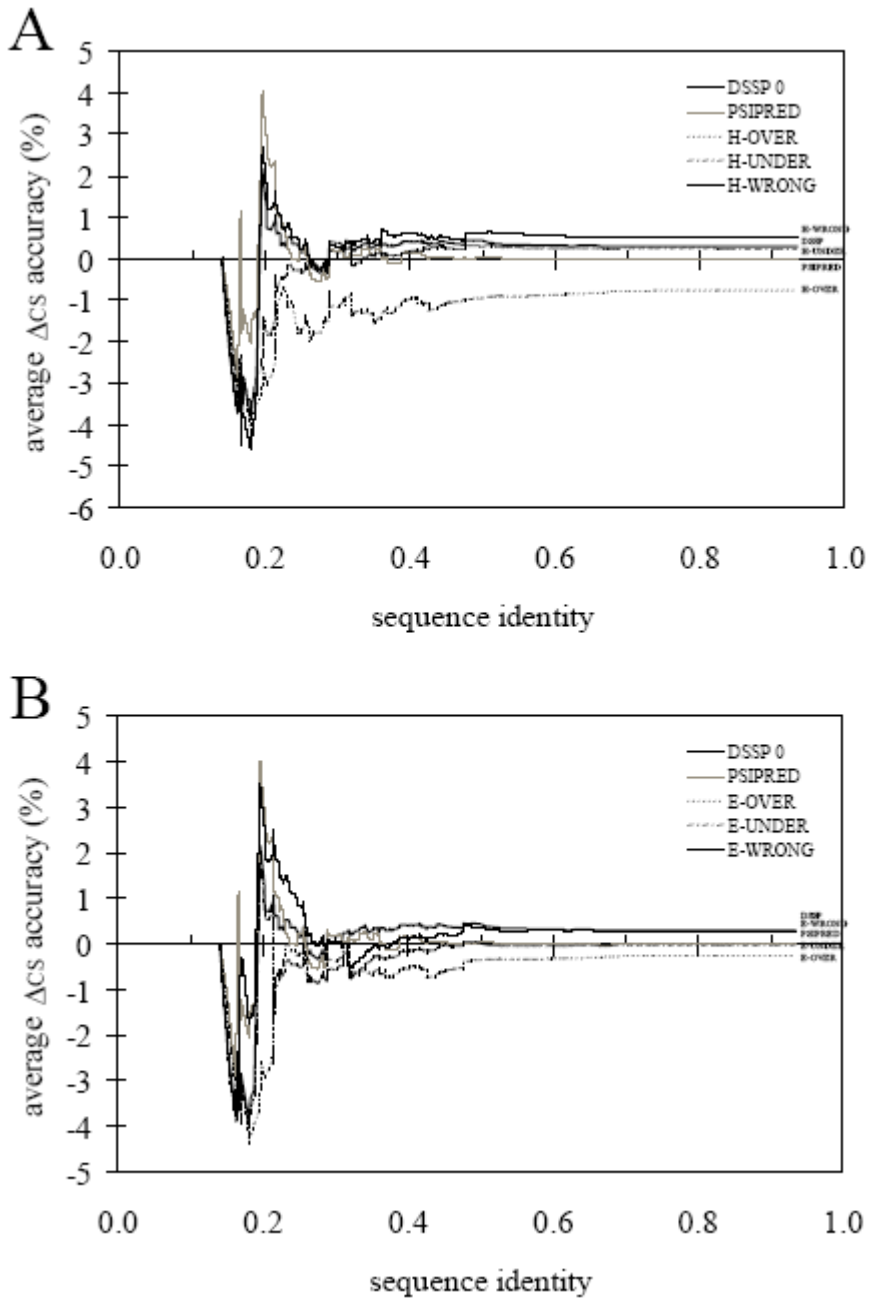


Figure. 7.5. The cumulative average delta (Δ) column (CS) score of the multiple alignments produced by $PRALINE_{PSI\text{-}SS}$ compared to those produced by the $PRALINE_{PSI}$ as a function of sequence identity. $PRALINE_{PSI\text{-}SS}$ is given DSSP (DSSP) and PSIPRED (PSIPRED) secondary structure information to guide the alignments. In addition, the DSSP assignments are modified with (A) helix-specific and (B) strand-specific errors. H-OVER and E-OVER correspond to alignments using DSSP secondary structure information where helix and strand over-predictions are maximized, respectively. H-UNDER and E-UNDER correspond to alignments using DSSP secondary structure information where helix and strand under-predictions are maximized, respectively. H-WRONG and E-WRONG correspond to alignments using DSSP secondary structure information where helix and strand switches are maximized, respectively. Each data point represents the average improvement of all data preceding it, i.e. the last data point represents the overall improvement.

Also, we have observed that averaging the secondary structure information over

the whole position of a homology-extended profile in the PRALINE$_{PSI}$ strategy improves the alignment quality more than 1% overall. By averaging we mean that all residue scores of that position become an un-weighted mixture of the helix, strand and coil content over all query sequences rather than a weighted mixture of the query sequence and added homologues. Although a better result, the reasons for this advantageous behaviour of the method are not entirely clear at this point. A possible explanation might be that the local alignments of the homologous sequences detected by PSI-BLAST has a noise level that warrants taking the overall query sequence distribution rather than that over all sequences (query and homologous sequences).

In fields of growing importance such as homology modelling and structure prediction, the use of high quality multiple alignments is a critical step. However, the limitations due to secondary structure quality reported in the literature and in this study still remain a problem in both the fold recognition and sequence alignment fields. Addressing these issues in secondary structure prediction should become a priority, as it would help lift the improvement capabilities of at least two of its related fields.

7.6. ACKNOWLEDGEMENTS

The authors thank the Vrije Universiteit Amsterdam for funding this project. Special thanks are also due to Dr. Jens Kleinjung for useful discussions and critical reading of the manuscript.

Chapter 8

The PRALINE Server

The content of this chapter has been published in Simossis VA, Heringa J (2003) The PRALINE online server: optimising progressive multiple alignment on the web. Comput Biol Chem 27:511-519 and Simossis VA, Heringa J (2005) PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. Nucleic Acids Res. (Server Issue 2005) (in press).

The PRALINE WWW server has been designed to provide both the non-specialist as well as the specialist users with a versatile toolbox to align protein sequences. To this end, the server can be used through two different interfaces: a standard user interface and an advanced user interface. We provide online help sections for each of the different parameters PRALINE may be set with, containing background information and examples.

The server can be used to run PRALINE with a variety of different alignment strategies: standard global progressive alignment (Heringa, 1999), global progressive alignment with global profile pre-processing (Heringa, 1999), global progressive alignment with local profile pre-processing (Heringa, 1999) and global progressive homology-extended alignment (Simossis et al., 2005). All these strategies can also optionally integrate predicted or "true" secondary structure information (Heringa, 1999, 2002; Simossis and Heringa, 2005a). Finally, the profile pre-processing strategies and the use of specific secondary structure prediction methods can also be optionally used to iteratively optimise the alignment (Heringa, 2002). These options are briefly summarized in the next sections, followed by a detailed description of the server interfaces and output.

8.1. PROFILE PRE-PROCESSING AND ITERATION

The profile pre-processing threshold values are alignment-dependant and therefore, it is up to the user to decide on an optimal value. All pair-wise scores are saved in a list that is available on the results page, after an initial run. This means that it would be sensible to run an alignment using a threshold value of 0, which will include all sequences, and then choose an optimal threshold value from the score list on the results page and re-run the alignment using that threshold value. If a negative threshold value x is used ($-x$), the threshold scores are weighted each time according to the sequence lengths; otherwise, the length is not taken into consideration. Heringa (2002) recommends a setting of -9.5 for the length-dependent threshold value.

In addition, the two profile pre-processing methods provide iteration capabilities from 0 to 10 iterations. Finally, when using a profile pre-processing method, PRALINE produces the alignment providing reliability scores for each amino acid position as well as an average reliability for each alignment position, at each iteration. The reliability

scores are represented in the position reliability colour scheme. PRALINE provides a number of alignment strategies such as profile pre-processing and iterative alignment optimisation (Heringa, 1999, 2002). The secondary structure-guided strategies using PHD, PROF, JNET and SSPRO and the profile pre-processing strategies can be set to use consistency information to drive subsequent alignment rounds (iterations), each time drawing upon the theoretically higher quality information from the previous cycle. A detailed account of these strategies can be found in previously published work (Heringa, 1999, 2000, 2002; Simossis and Heringa, 2003; Simossis et al., 2003; Simossis and Heringa, 2004b; Simossis and Heringa, 2005a; Simossis et al., 2005).

8.2. HOMOLOGY-EXTENDED MULTIPLE ALIGNMENT

When used as an option on the server, the homology-extended alignment strategy (see Chapter 6) can be further customised by manually entering the desired iteration count, starting e-value cut-off and database to be searched by PSI-BLAST for the building of the homology-extended profiles (default: 3 iterations, starting with a cut-off of $10e^{-6}$ on the NR database). The default parameters have been optimised by testing different settings on the HOMSTRAD database of structural alignments (Simossis et al., 2005).

8.3. INTEGRATION OF SECONDARY STRUCTURE

The secondary structure integration options of PRALINE (see Chapter 7) involve using any one from a list of seven prediction methods PHDpsi (Przybylski and Rost, 2002), PROFsec (Rost, personal communication), SSPRO 2.01 (Pollastri et al., 2002), YASPIN (Lin et al., 2005) (also see Chapter 5), PSIPRED (Jones, 1999), JNET (Cuff and Barton, 2000) and PREDATOR (Frishman and Argos, 1996, 1997)) to predict the secondary structure of the input sequences. In addition, the user can optionally select to also search the PDB to find 3D structure information for the input sequences and use the DSSP derived secondary structure for the alignment. If both DSSP and a prediction method are selected, the predictions will only be integrated into the alignment for those sequences that do not have a PDB entry. Finally, in the same list as the seven prediction methods, an optimally segmented (Simossis and Heringa,

2004a) (also see Chapter 4) or majority voting consensus can be alternatively used that currently combines the predictions of PROFsec, SSPRO and PSIPRED.

8.4. THE PRALINE SERVER

The PRALINE program is designed to use two or more input protein sequences in FASTA format (Pearson, 2000). The proposed maximum number of sequences that should be submitted to the server is set to 500 with length 2000, but this is mainly to limit the server load and is not the PRALINE program's limit. Also, due to the long running time needed for strategies such as $\text{PRALINE}_{\text{PSI}}$, an optional e-mail notification can be requested that is delivered upon a job's completion and contains the link to the results and some statistics on the resulting alignment. PRALINE can be run using its default settings (gap opening penalty 12.0, gap extension penalty 1.0 and the amino acid substitution matrix BLOSUM62, to do a single global alignment of the sequences) or otherwise, there is a help section to describe how the gap penalties work and some example combinations for standard amino acid substitution matrices. At present, the amino acid substitution matrices available are PAM250, BLOSUM50, BLOSUM62 (Dayhoff et al., 1983), and GON250 (Gonnet et al., 1992).

8.4.1. The standard user interface

The standard user interface is targeted mainly towards non-specialist users (Figure 8.1). The alignment strategies and optional outputs are listed as selectable options. The user can alter the gap penalties, select different matrices and set related alignment strategy parameters if needed. Although all the main PRALINE parameters are available as selectable options, not all are made available through this interface because many of them are too specialised. For users that want to make full use of all the PRALINE parameters we have created the advanced user interface described next.

8.4.2. The advanced user interface

Instead of selectable options, the advanced user interface has a command line so that the user can manually enter more options than provided in the standard interface (Figure 8.2A). In addition, we provide the user with the ability to use a custom amino acid substitution matrix that can be uploaded for use in the same way as an input

sequence file (Figure 8.2B). A sample amino acid substitution matrix is made available for viewing in the format that PRALINE can read it in. Finally, the user can use the reference options table that has all the options currently available to PRALINE with a short description of each option (Figure 8.2C).

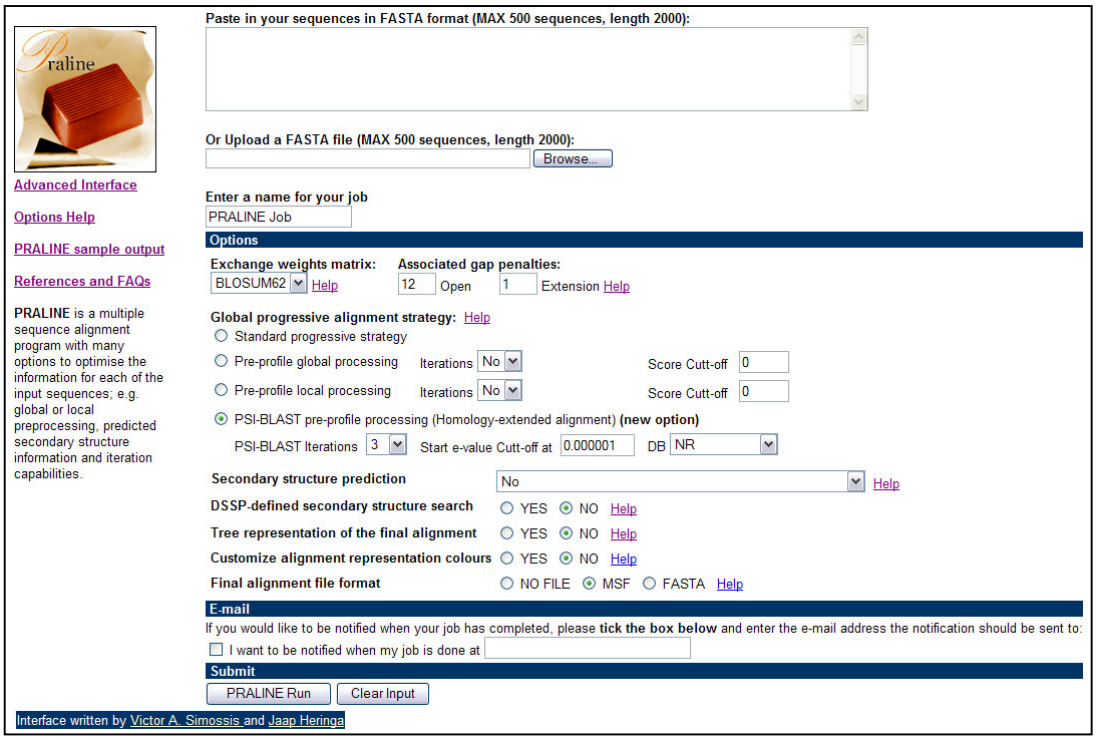


Figure 8.1. The PRALINE Server standard user interface. a) Text area for FASTA or PIR sequences b) path for uploading a FASTA or PIR file, c) submit job for default run, d) gap penalties and amino acid exchange weights matrix selection, e) alignment method selection, f) secondary structure information (no iteration at present), g) select tree representation, h) select user-defined colour scheme, i) select final alignment file format.

8.5. THE OUTPUT PAGE

The output page is automatically displayed once a job is complete and contains various parts depending on the options selected (Figure 8.3). In order to provide all generated files for the user to keep there is a link to download a compressed file with all the results in the job directory (Figure 8.3D) and also individual links that allow the user to download specific files related to each sequence in the set (e.g. a PSI-BLAST profile or a secondary structure file) (Figure 8.3E).

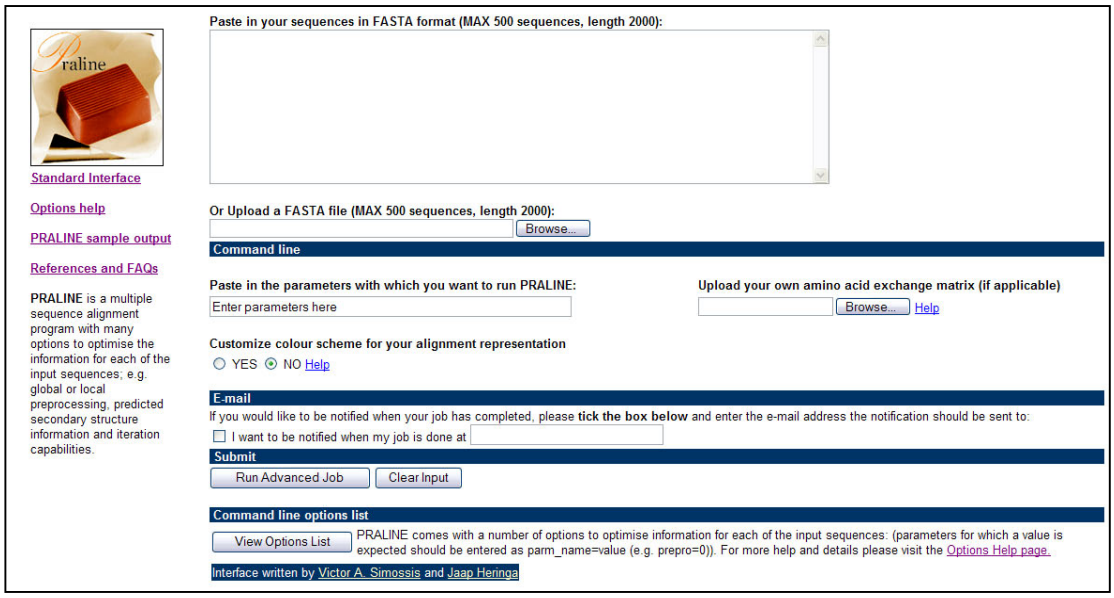


Figure 8.2. The PRALINE Server advanced user interface. a) Text area for FASTA or PIR sequences b) path for uploading a FASTA or PIR file, c) command line for PRALINE options, d) path for uploading user-defined amino acid exchange weights matrix, e) select user-defined colour scheme, f) complete PRALINE options list.

If the iteration number selected is greater than 0, a subtitle informs the user which iteration cycle results are presented on the page (Figure 8.3A). The alignment from each iteration cycle is presented on a different page and is accessible by the corresponding links (Figure 8.3C). In addition, it informs the user of the total time taken for the process to complete, provides statistics related to the visible alignment (Figure 8.3B) and if the iterations halted due to alignment convergence or limit cycle convergence and which iteration was the last (not applicable in the Figure 8.3 example). In the case of iteration-specific output such as that iteration's alignment or secondary structure prediction additional links are displayed (Figure 8.3F).

If profile pre-processing is selected the user has the option of viewing the profile pre-processing scores for all pair-wise alignments for deriving an optimum cut-off value (Figure 8.3G).

Finally, depending on the selected parameters of the job, a series of buttons allows switching between the available colour-coded or dendrogram views (Figure 8.3H) (details about the colour schemes are described in the next section). The dendrogram view only represents the hierarchical clustering of the progressive alignment. At any point, the visible alignment can be converted into a PDF for printing or further manipulation (Figure 8.3I). The remaining of the results page consists of a short description of the visible colour scheme with a key to the colours, after which the colour-coded alignment or dendrogram follows (examples of the available colour schemes is shown in Figure 8.4).

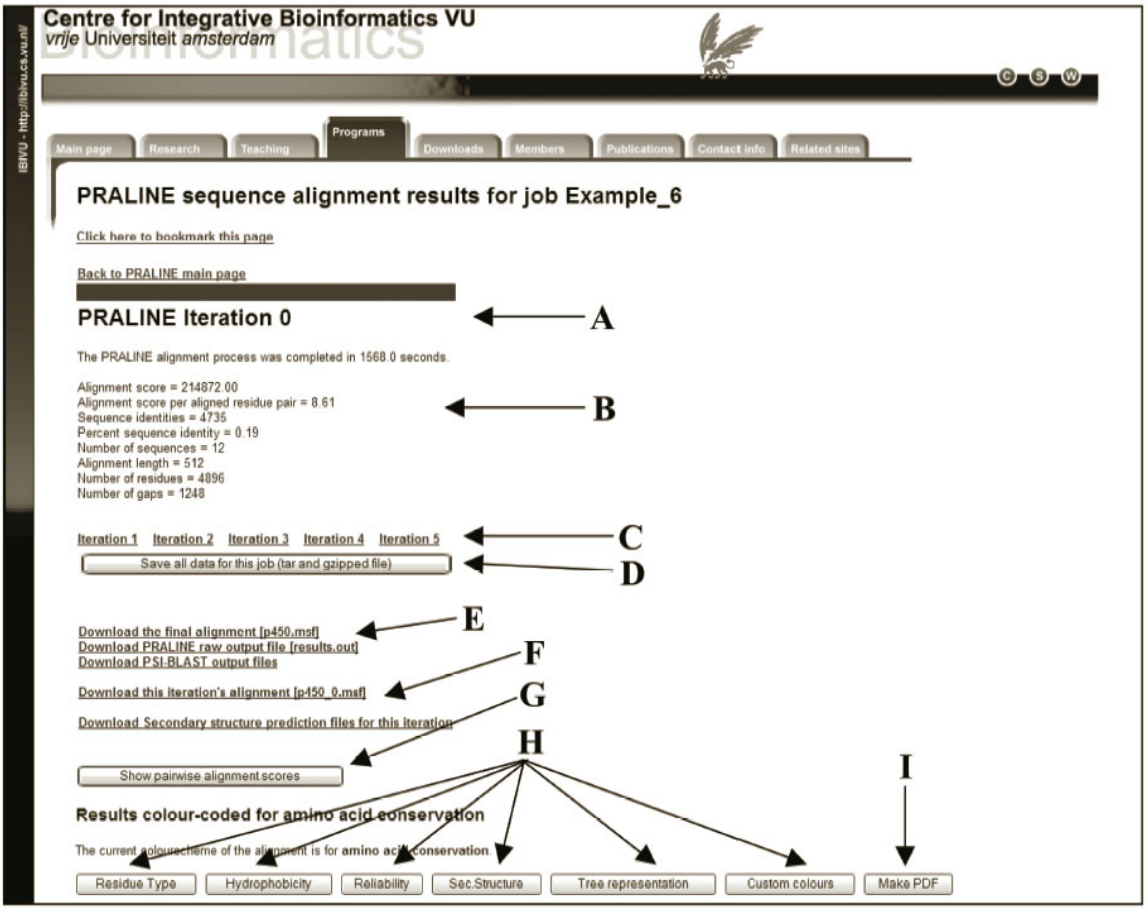


Figure 8.3. The PRALINE output page headers. A) The subtitle indicating which iteration results are presented on this page (only available if iteration>0 is selected), B) The time taken to run the job, C) The links to all other available iteration cycle results (only available if iteration>0 is selected), D) The link to download all job files as a compressed file, E) Links to tabulated specific file types, F) Links to iteration-specific output files (only available if iteration>0 is selected), G) The button that hides/ reveals the profile pre-processing scores of the sequence set (only available if profile pre-processing is selected), H) The buttons that switch between colour schemes, I) The button that generates and opens a PDF version of the alignment in the visible colour scheme.

8.6. COLOUR SCHEMES

The currently available colour schemes are based on residue type, conservation by alignment position, reliability by alignment position and position average reliability, hydrophobicity and finally secondary structure (Figure 8.4). Each scheme has a short explanation of how to interpret the colours and also a colour reference key at the top of the alignment. The default representation is the conservation scheme. Residue specific colours have been used in accordance with the colouring scheme of ClustalX (Thompson et al., 1997) and hydrophobicity scaling has been assigned according to (Eisenberg et al., 1984). The reliability colours are only available if profile pre-processing methods have been used. The secondary structure representation is in three states (H-helix, E-strand and blank-other). It is only available if secondary structure has been used to guide the alignment.

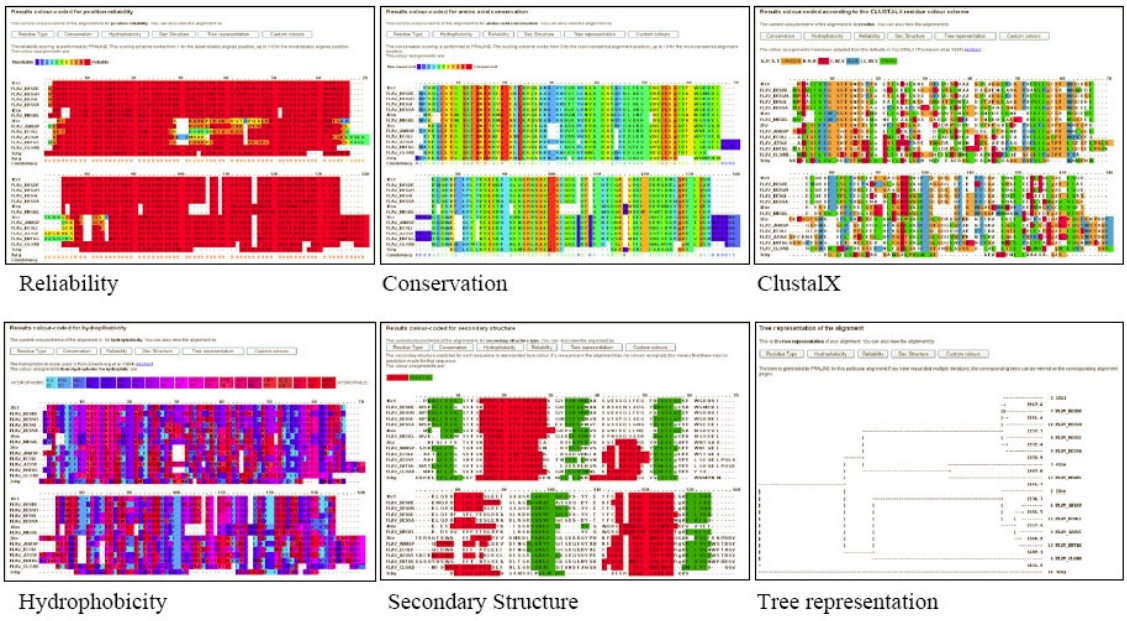


Figure 8.4. The alignment colour schemes. Position reliability, Conservation, ClustalX, Hydrophobicity and Predicted secondary structure. The other possible representation is the Tree representation.

Apart from the five pre-set colour schemes, we also provide a user-defined colour scheme. The user-defined colour scheme is optional. It enables the user to select from a table of eight pre-set colours and assign any of them to one or more amino acids, in any combination desired for viewing (Figure 8.5). This is particularly useful when a

specific position or a motif needs to stand out in the alignment, or if specific amino acids need to be depicted for illustrative purposes.

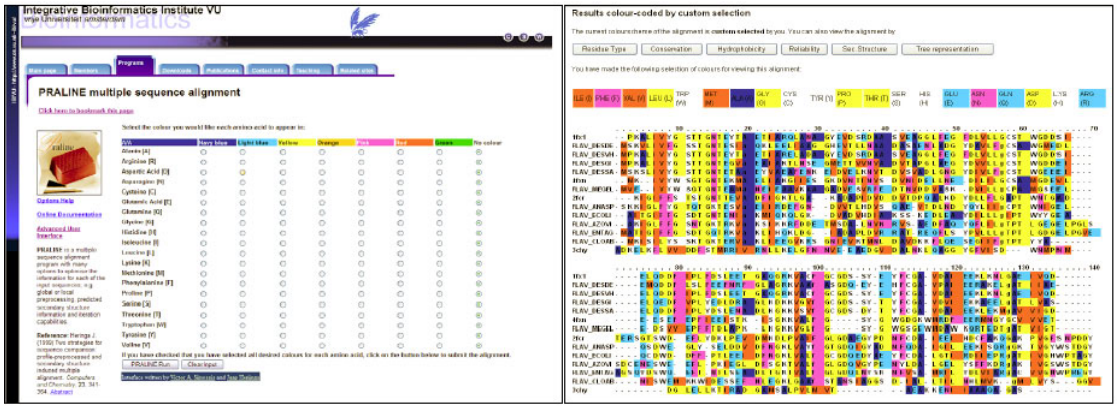


Figure 8.5. The user-defined colour table and a sample custom colour alignment representation.

8.7. SUPPLEMENTARY MATERIAL

Due to the large number of possible outputs, we have provided a set of 9 representative sample outputs for the "p450" HOMSTRAD sequence set (21% average sequence identity) alignment on the server, each one representing a different combination of PRALINE strategies and settings. These examples are intended as supplementary material to this article and can be accessed through a dedicated link on the server pages or directly at http://ibivu.cs.vu.nl/programs/pralinewww/example/.

In Figure 8.6 we illustrate sections of the PRALINE$_{PSI}$ alignment of the "p450" sequence set using both DSSP (Kabsch and Sander, 1983) and PROFsec secondary structure integration settings. The colour schemes in the figure are for positional conservation and secondary structure. The secondary structure information for each sequence in this alignment has been derived by DSSP, since all the sequences have a corresponding PDB structure.

The cytochrome P450 enzymes primarily act as oxidases in multi-component electron transport chains to break down naturally occurring toxins and mutagens. The structure is almost triangular, with the C-terminal part being mostly helical, while the N-terminal part is more beta sheet rich. The *signature motif* of P450 enzymes is the haem-binding site, which is often represented as FxxGxxxCxG (Figure 8.6C). Other

conserved regions include the motif A(A/G)x(E/D)T (Figure 8.6A) where the threonine

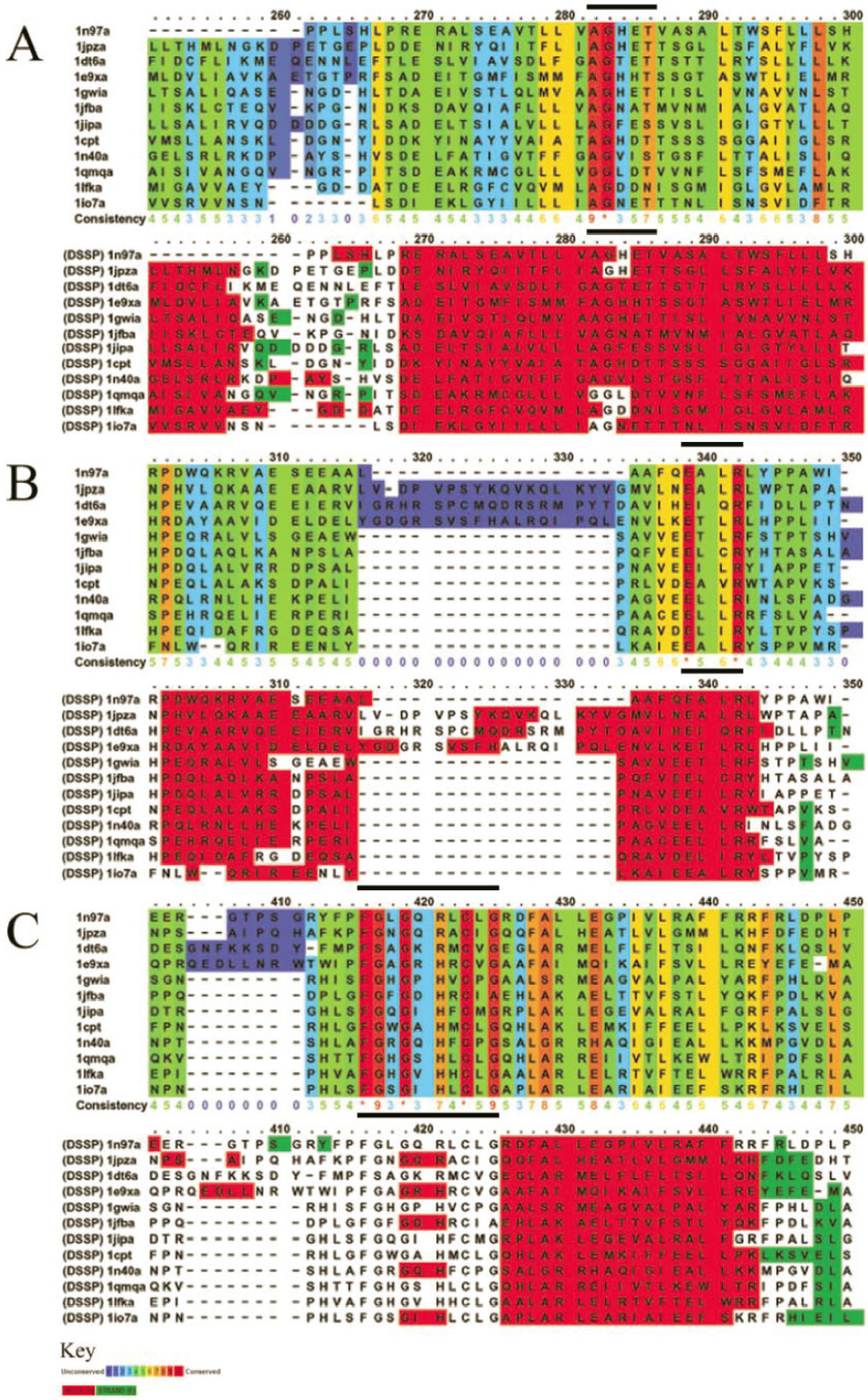


Figure 8.6. The PRALINE$_{PSI}$ P450 alignment using both PROFsec and DSSP secondary structure integration settings. The alignment has been sectioned to focus on the regions containing the conserved motifs of the cytochrome p450 enzymes (signified by the black bars above the rulers). A) The oxygen-binding motif, B) the ExxR motif and C) the haem-binding motif. For each section, the top colour scheme shows conservation levels according to the colour key and the bottom one shows the secondary structure each residue belongs to (red: helix, green: strand, clear: coil). The ruler on top of each alignment block shows which parts of the alignment are visible.

(T) residue is part of the oxygen binding site and an invariant ExxR sequence (Figure 8.6B). The ExxR and the C residue at the haem-binding site are the only completely conserved amino acids in P450s. These well-documented details are straightforwardly visualised in the PRALINE output conservation colour scheme, while the secondary structure view allows us to relate them in a structural context. As stated in the literature (Ortiz de Montellano, 1995), the oxygen binding and ExxR motifs are each part of two distinct C-terminal helices, while the haem-binding motif flanks the N-terminal end of the last helix. Due to space limitations the alignment has been sectioned to concentrate on these regions, but the full alignment can be viewed online in example 9 of the supplementary material.

8.8. CAVEATS

The PRALINE Server has some limitations that need to be clear to the user. Firstly, PRALINE is not a DNA alignment program and does not accept DNA sequence as an input, nor does it translate it into protein. Secondly, profile pre-processing, secondary structure prediction and iterations make a huge improvement in alignment quality and information feedback, but can make PRALINE slow, albeit a parallelised version has made available (Kleinjung et al., 2002). Finally, all alignment methods will produce some sort of alignment whether biologically meaningful or not. However, the ability to manually optimise parameters and the position reliability scores provided by PRALINE allow the user to make a reasonable assessment of the alignment quality and choose the best resulting alignment.

8.9. CONCLUDING REMARKS

The PRALINE server offers some unique features that make it a versatile and useful alignment tool. It provides the user with feedback about the quality of the alignment produced in an iterative scenario and in addition enables the user to use this information to optimise the alignment by having fully customisable parameters. Another feature is that it provides more than one alignment strategy and can use secondary structure input, thus covering a wide range of alignment cases. In addition, the multiple representations of the alignment offer a convenient and diverse way for

alignment illustration according to the users needs. Apart from being an accurate method, the PRALINE Server is a toolbox for protein sequence alignment that gives users the opportunity to learn more about their alignment problem, the means to find a best possible solution and present it in a more detailed and educational form.

8.10. ACKNOWLEDGEMENTS

The authors would like to thank the Vrije Universiteit Amsterdam for funding this project. Special thanks are also due to Dr. Franca Fraternali, Dr. Jens Kleinjung and Dr. John Romein for help with debugging and server testing.

Chapter 9

General Discussion

A MSA can be viewed as an inflexible representation to obtain a unified picture of the relatedness of a set of sequences by averaging out matched residues that possibly cannot be consistently matched over the entire length of the sequences. This is because evolution, through mutations, insertions and deletions of sequence fragments, works on spatially and temporary de-coupled molecules, so that sequence alignment incompatibilities can well arise under divergent evolution.

Given these complications, building a reliable MSA for a query set of sequences is a daunting task. In this thesis it has been made clear that the increased attention for multiple sequence alignment methodology have resulted in recent developments regarding most of its facets. Computational issues have been addressed both by adapting methods to high-throughput computing by code parallelisation and by new speed-optimised alignment formalisms, such as the recent methods POA (Lee et al., 2002) and MUSCLE (Edgar, 2004). Sensitivity has been increased by the development of enhanced techniques for carrying out simultaneous alignment, by devising new profile formalisms, by combining local and global alignment, by new iterative schemes, and by the emergence of new schemes to score and exploit the consistency of alignments.

The increased focus has also led to the construction of new benchmark databases and novel evaluation protocols. Further developments will be crucially dependent on the integration and representation of biological knowledge in new quality criteria. There is now a multitude of high-quality MSA techniques, each of which with particular strengths and weaknesses. However, no single best method exists, as each method will have an alignment case on which it performs best. Increased sensitivity could abound from new consensus protocols to utilise the combined power of the techniques, or from new techniques to determine the kind of alignment problem at hand and then to subsequently invoke the most appropriate method or combination of methods available. In the meantime, it remains important for the end-user to run a combination of different MSA methods to optimise the biological information derived from a set of sequences, either through visual inspection of the resulting MSAs or by the application of other bioinformatics techniques that use these MSAs as input.

The research presented in this thesis has explored new ways to improve secondary structure prediction and multiple sequence alignment, individually and also

as part of a co-operative schema. In addition, as a result of the research performed during this project, a number of very useful applications has been developed, which are increasingly gaining in popularity in the research community. This final chapter summarises the main results and conclusions of this research and as such, addresses the questions raised by the initial design of the project (see Chapter 1). In addition, new questions for future research are presented for the further improvement of the increasingly merging fields of multiple sequence alignment and secondary structure prediction, as well as extensions to other related fields.

9.1. REVIEWING THE KEY RESEARCH QUESTIONS

9.1.1. Can the use of alignment information for secondary structure prediction be improved?

As one of our first investigations, we addressed the issue of alignment information usage by secondary structure prediction methods. In Chapter 3, an up-to-date overview of the secondary structure prediction field has been presented, stressing the fact that the prediction process of most methods has become a standardized series of steps with the only main difference being the machine learning approach applied. In most cases, the local alignments generated by a database-searching tool (e.g. PSI-BLAST) are used blindly both for training and eventually as input for the prediction algorithms. Admittedly, the resulting increase in position-specific information content is one of the main innovations that have benefited prediction accuracy (Przybylski and Rost, 2002). However, improving the quality of the alignments produced by the database searching tools would add even more to the benefit and although this point was not addressed in this thesis, it is an aspect that will be considered in future work.

The sensitivity of secondary structure prediction to the input alignment quality is a well-documented fact and therefore the blind use of the low quality local alignments produced by database searching tools is merely a compromise to reduce the time taken to do a prediction, albeit at the expense of quality. However, this loss of prediction quality is not dramatic and does not apply to all cases since not all alignments are of low quality and the predictions on average are more than 75% correct. Nonetheless, as we have shown in Chapter 7, even high accuracy predictions

contain errors that are an important limiting factor for possible enhancements in related fields such as multiple sequence alignment.

Another inherent flawed use of the alignment information is that some prediction methods remove information from the input alignments to facilitate the information use by the machine-learning strategies. In particular, the alignment positions with a gap in the top sequence are completely removed prior to prediction by many methods, as discussed in Chapter 4. By re-introducing as much information as possible back into the predictions as a post-processing step, we were able to show that in many cases, the additional information improved prediction quality even in state-of-the-art methods such as SSPRO (Pollastri *et al.*, 2002) and JNET (Cuff and Barton, 2000). However, these methods have been optimised to use local alignments generated from database searching tools. It would therefore be interesting to see what the resulting improvements in secondary structure prediction would be if instead of model alignments (like the ones used in Chapter 4), local alignments generated by database searching tools were optimised by the suggested post-processing technique.

The analysis of different secondary structure prediction methods in Chapter 4 also showed that the majority of prediction errors are concentrated in the edge regions of secondary structure elements, while the central regions are mostly assigned correctly. Based on this, we developed the YASPIN secondary structure prediction method (Lin *et al.*, 2005), which we have presented in Chapter 5. YASPIN has introduced two major innovations to secondary structure prediction: firstly, it classifies amino acids into seven secondary structure classes by discriminating edge and central secondary structure element positions (Hb, H, He, Eb, E, Ee, C), instead of grouping them all into the traditional three (H, E, C); and secondly it combines two separate machine learning techniques for the prediction. Although the secondary structure prediction accuracy of the method as measured through Q3 and SOV (Zemla *et al.*, 1999) scores, did not surpass the current state-of-the-art methods, YASPIN's ability to correctly predict strand, the hardest class to predict, was significantly higher than that of any other method.

In conclusion, our research has shown that the use of alignment information by secondary structure prediction methods can be improved in at least three ways. It would be interesting to see if these improvement strategies could be applied to state-of-the art

methods in order to reduce the amount of prediction errors that seem to affect many important related fields.

9.1.2. Can prediction errors be limited by optimally combining the best predictions from a number of available state-of-the-art methods?

Although the way alignment information is used in secondary structure prediction is unquestionably very important for prediction quality, the training of the machine-learning method is equally responsible for how accurately a prediction method classifies residues into secondary structure classes. One problem with training a machine-learning method is the choice of training set. Conveniently, large initiatives have made the standardisation of the training sets possible by keeping up-to-date databases of all currently available structures and their family classification. As a result, using the same training sets, the various machine learning methods can be more objectively tested for there ability to "learn" and apply their classification power to an unknown query. On the other hand, as has been repeatedly shown in our research, current sate-of-the-art prediction methods have very high correlation between their predictions, albeit there are still some cases where the prediction accuracy varies significantly between methods.

In the past, the extent of variation in assignments between top-reforming methods was high enough to be used to reduce the biases of each individual method by generating consensus predictions by *majority voting* (MV). In the majority of studies involving benchmarks of secondary structure prediction methods it is common practice to derive a consensus prediction from the top performing methods, resulting in slightly better predictions than attained by any of the individual methods involved. A question that can be asked is whether the development cycle is such that secondary structure predictions should converge to a consensus form until a new strategy is designed that performs even better. And what would the consensus be like if this new hypothetical method was then used together with other top performers? In any event, the main drawback of consensus methods is the time the researcher has to invest, as more than one method has to be run, while some of the methods are not freely available. Unfortunately, the latter is a separate problem in itself irrespective of consensus prediction.

Having developed the YASPIN method (Chapter 5) that showed a superior ability to predict strand we tried to introduce this higher quality strand information into a consensus with other high-performing prediction methods. Also, in addition to the MV technique we applied the optimal segmentation dynamic programming (DP) consensus scheme originally introduced in Chapter 4. The results of this research were not submitted for publication before the submission of this thesis and have therefore not been included as one of the chapters. However, as mentioned earlier, the results from this research show that methods appear to perform very similarly overall, thus limiting the possible improvement by consensus prediction derivation, contrary to the observed results of older studies of this type. This suggests that prediction methods are nearing an upper limit and a new "edge" is needed to surpass the 80% accuracy upper margin, as it happened in the early 1990's when the 60% accuracy margin was overcome after a long uneventful period for the secondary structure prediction field. The use of long-range interactions, residue contacts and the use of predicted tertiary structure information have already shown promising results and possibly the effectiveness of consensus predictions will become significant again when the new age of prediction methods make their appearance.

9.1.3. Can the collection of homologous information from sequence databases improve accuracy of multiple sequence alignment as has been shown for secondary structure prediction?

The alignment of sequences depends on an evolutionary model, while the prediction of secondary structure is more a pattern recognition problem where correlated amino acid propensities delegate low-level structural element classes. Nonetheless, although these two fields are quite different, the use of additional homologous information from database searching has been instrumental in the improvement of both of them. In Chapter 6 we described a homology-extended multiple sequence alignment algorithm that uses profiles generated by database searching instead of the query sequences in a given set. The clear higher multiple alignment quality shows that a good representation of the evolutionary history of a sequence or a set of sequences allows a sensitive and accurate detection of additional homologous sequences by pair-wise comparison and further improves the alignment of

multiple sequences. In addition, these higher quality alignments also improve the quality of the predicted secondary structure of the sequences.

This raises several questions as to what the common connection is between the amount of homologous information available and the performance of these different fields. Also, what other specialised innovations could be tested that may result in beneficial results? For example, we previously discussed the possible better alignment of the sequences detected by database search tools for secondary structure prediction. Similarly, this would also benefit the homology-extended alignment strategy since it is dependent on the accuracy of the position-specific information of the homology-extended profiles.

9.1.4. Could the simultaneous use of extended homologous information and the resulting secondary structure predictions lead to an additive improvement?

The combination of extended homologous information and an anchoring template such as secondary structure to guide the alignment of distant protein sequences is very appealing. However, as we have discovered from combining the two in the alignment strategy described in Chapter 7, the resulting benefits although significant, are lower than one would expect. Ultimately, it appears that the use of extended homologous information is more beneficial than simply using secondary structure to anchor the alignment of distant sequences. This is not surprising since protein secondary structures tend to vary a lot between very distant relatives, i.e. <20% sequence identity, and may consequently induce erroneous biases into the alignment scoring calculations. However, at this stage of our research the predicted or “true” secondary structures of the query sequences have been applied as common structural information for all other sequences in the homology-extended profiles, leading to a great generalization of the information as we have discussed in the concluding remarks of Chapter 7. It is enticing to consider a scenario where known secondary structure information is incorporated for the individual members of the homology-extended profiles, where available, and also any other useful information such as domain and functional motif annotations, which are increasingly being integrated into databases that are freely available. This way, we not only provide the alignment method with more robust anchor regions to guide the alignment, but also we may be able to correct

local alignment errors that the database searching tools make when building the profiles.

Considering the improvement in alignment accuracy made possible by simply using the sequence information in the homology-extended profiles (Chapter 6) and the additional benefit observed even when using a generalised form of predicted secondary structure information (Chapter 7) suggests that the integration of heterogeneous information into alignment profiles is a promising area to investigate.

9.1.5. What are the types of errors in predicted secondary structure that limit multiple alignment improvement capabilities?

In more than one occasion during this research (Chapters 4 and 7) we have come across limitations in the ability of secondary structure prediction methods to correctly guide the alignment of sequences. Despite the high per-residue accuracy of these methods, specific prediction errors seem to be the source of these limitations compared to that possible when using the "true" information. Although in general the detection of core regions of secondary structure elements is reliable and beneficial to alignments, it appears that the edge regions also contribute to a certain extent.

As we have discussed in the concluding remarks of Chapter 7, this should direct the attention of the secondary structure prediction field to addressing these regions more carefully in future prediction methods. In our research for the YASPIN method (Chapter 5), we made an attempt to classify the edge regions separately to the core as discussed earlier in this discussion. Considering the increased use of support vector machine (SVM) technology and its advantages over other machine learning techniques for pattern recognition, it should be interesting to design a method that not only classified residues into the commonly used 3-class alphabet, but rather identify helix- and strand-specific edge classes and train a group of SVMs to differentiate them from the core elements before combining the predictions for different classifications into a single final prediction.

9.1.6. Alignments affect secondary structure prediction accuracy and *visa versa*: What factors limit a smooth interdependence?

Although this aspect was not directly addressed in the research, a few key points can be raised from the work presented in this thesis and from preliminary work done on iterative alignment-prediction mutual optimisation. Having designed a secondary structure-guided alignment algorithm for PRALINE (Chapter 7) we performed some preliminary tests using the PHD (Rost and Sander, 1993) secondary structure prediction method. The alignment strategy proceeds by initially constructing a MSA without information about the corresponding secondary structure and is used as input to PHD. Then, the predicted secondary structures are incorporated to produce a theoretically better MSA. In turn, a new secondary structure prediction is performed using the "new" secondary structure-guided MSA. This inter-dependence allows for an iterative scheme where each iteration proceeds in the same way as described above, producing a MSA that is passed on to the next iteration and guides secondary structure prediction, which in turn guides alignment and so on. In Figure 9.1 we show the PHD predictions for the orphan member of a set of 14 flavodoxin proteins (for evolutionary tree see Figure 2.3 in Chapter 2), over 10 iteration cycles. The original prediction (INIT) that is based on the normal progressive alignment of the 14 flavodoxins has obviously made critical mistakes (regions in grey boxes) and has even missed out a whole strand element in the C-terminal part of the protein. In the successive iteration rounds the use of the newly predicted secondary structure improves the MSA quality to such a point that the PHD method can immediately correct errors made (INIT vs. ITER 1) and through gradual optimisation also detects signals that were not evident before iteration 4. This is extremely promising because the cheY sequence is the most distant member of 14 flavodoxins making it the hardest to predict when only using the other 13 more closely related members as related information for PHD. However, there are regions in the prediction that do not behave as successfully, such as the periodic detection or complete loss of the second strand element. Admittedly, the use of PHD is at present outdated and more accurate methods exist to perform the predictions that also respect the alignment such as PROFsec, SSPRO and JNET, so these are also integrated into the PRALINE method for this purpose.

In order to achieve a smooth inter-dependence between the residue and secondary structure components of this iterative scheme, attention to critical prediction errors and a way to identify high quality predictions and alignments through the

iterations is needed in order to guarantee that the best possible result is obtained with the minimum number of iteration cycles. To achieve this, we need to make use of database searching techniques for maximising the information related to each of the sequences in a set and at the same time monitor the resulting prediction and alignment quality using robust assessment measures at each iteration.

The important observation here is that iterative optimisation of an alignment not only based on positional consistency, but based on external information (homologous information, secondary structure) that itself is optimised along the way shows very promising preliminary results. Much like the other questions and possible extensions suggested in this discussion, we propose this as a next step from the work presented in this thesis and we elaborate on it further in the next section.

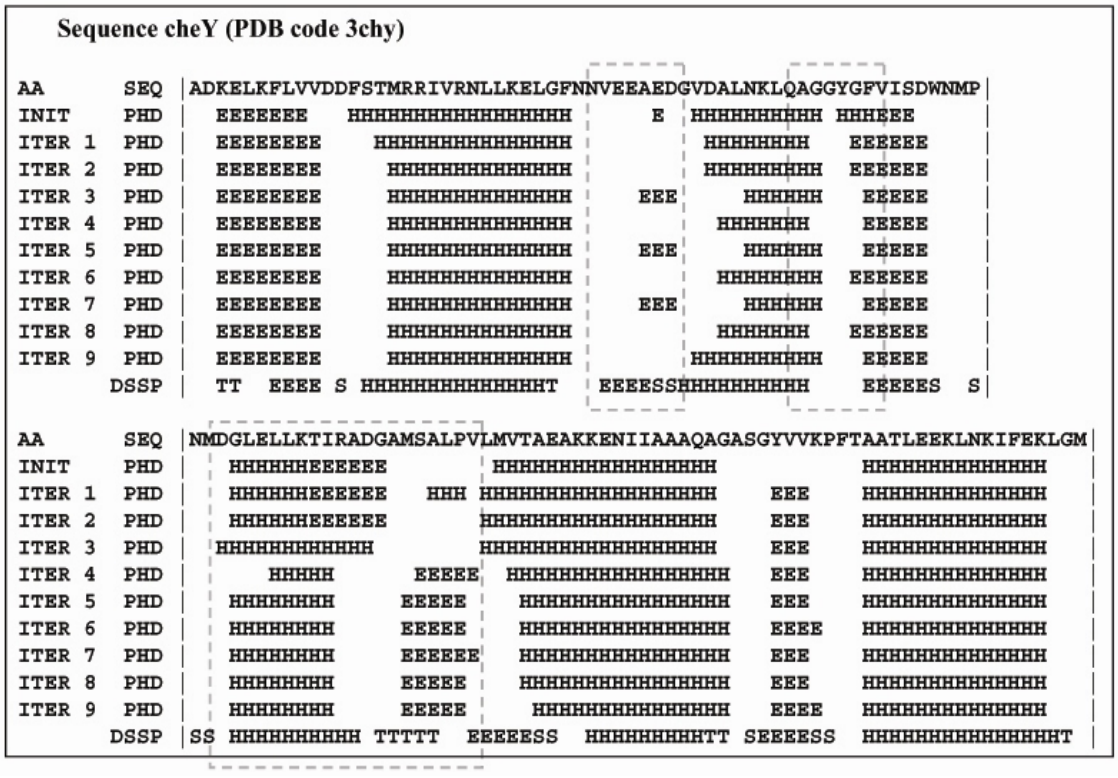


Figure 9.1. The inter-dependence of MSA and secondary structure prediction quality. The AA row represents the top sequence (cheY) of a MSA of 13 flavodoxin sequences. The INIT row is the original prediction produced by PHD on a simple dynamic programming alignment of the sequences. The numbered ITER rows show the influence of iterative secondary structure-guided optimisation of the same alignment using the PRALINE MSA method (see MSA section), on the prediction accuracy of PHD. The DSSP row is the DSSP-derived secondary structure of cheY from the PDB structural information. The regions boxed in grey dotted lines show areas where an increase in MSA quality induces improvements in the accuracy of the predicted secondary structure.

9.1.7. Is the inter-dependence of multiple sequence alignment and secondary structure prediction a key aspect for designing a mutual optimisation scheme?

The combination of MSA techniques and secondary structure prediction has come of age. Since the first use of MSA in secondary structure prediction, the process has evolved into a network of inter-dependant strategies that have become inseparable. First, a biologically meaningful set of sequences needs to be selected from sequence databases; then, these sequences need to be aligned in a biologically correct way either by the search method itself or by a separate MSA method. As the accuracy of the secondary structure prediction depends critically on the quality of the input MSA, it can be revealing to attempt various MSA methods. Selecting a representative set of homologous sequences that covers the amino acid variability at each alignment position is very important for optimal MSA. Consequently for secondary structure prediction correctness since the evolutionary information exhibited by MSAs provides the basis for secondary structure prediction.

The current secondary structure prediction servers have mostly integrated the three main components (database sequence searching, MSA and secondary structure prediction) at their respective sites. Generally, the servers have a locally running version of PSI-BLAST or SAM; they use the resulting MSA in the required format or even re-align the sequences using a separate MSA method; and they predict the secondary structure either using a single method or derive a consensus from a number of predictions by separate methods. However, more developments can be expected in the near future from the integration of the three main steps by possible iterative optimisation scenarios at different levels. In Figure 9.2 we show some iteration possibilities based on the inter-depending functionality of the three components. The iterative optimisation of MSA and secondary structure prediction, now implemented in for example the PRALINE method, can result in an optimised output for both the MSA and the secondary structure prediction. This can then be used to iteratively optimise the initial sequence search. Before each iteration cycle, the input MSA template can be filtered using the sequence- and structure-based scores of each sequence originally included in the sequence set, such that low scoring sequences are removed from the template. The resulting optimised sequence set can then lead to optimised MSA information, which in turn will allow more accurate sequence selection and more

correct detection of distantly related homologues from the sequence databases.

A long-standing idea has been that secondary structure prediction would not be able to go beyond 80% accuracy because the methods do not take long-range interactions into account; i.e., spatial interactions between residues that are separate in the sequence. Some methods such as PREDATOR and SSPRO have made an attempt to incorporate long-range interactions. It is likely that further development of computational methods exploiting such integrative strategies will elevate secondary structure prediction beyond the currently unreachable limit of 80% prediction accuracy.

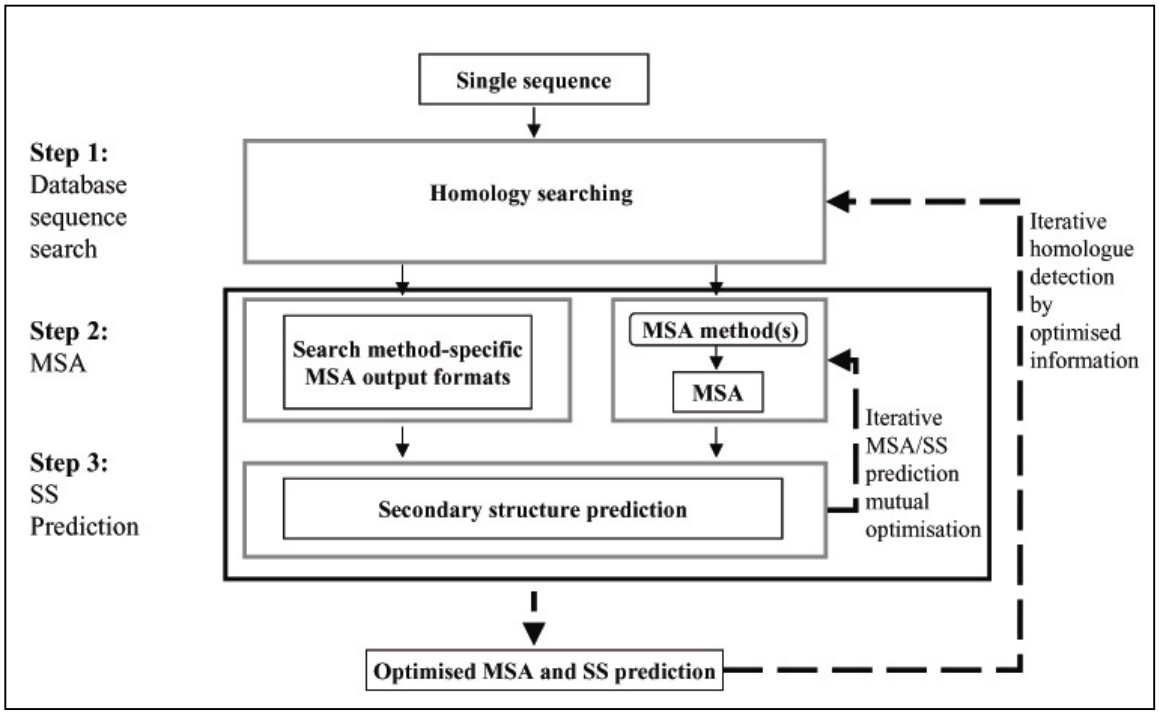


Figure 9.2. Iterative optimisation possibilities at different levels of the three-step secondary structure prediction process. First, iterative optimisation of the MSA using increasingly better secondary structure predictions. Second, iterative optimisation of the sequence selection and filtering of homologous sequences by using the optimised MSA information from each previous iteration as a guide.

9.2. EPILOGUE

The cornerstone of many modern biological research areas that is multiple sequence alignment, influences many other fields and as such creates opportunities for improvement in down-the-line research areas. In this discussion, many follow-on

projects and areas that need to be focused on in future work have been suggested. Beyond these, there are also application pipelines that the PRALINE alignment method could be incorporated into, such as genomic and proteomic annotation initiatives and protein structure and function analysis servers.

Postface

Finally, at the end of this thesis I would like to close by quoting a passage from a book that has influenced me greatly since I was very young. Much like the way I chose the title for this thesis, I have found inspiration in the words of a British professor and author who never seemed to want to do things the way they were supposed to be done, but found innovating creativity in attempting the seemingly unconventional. In this ending passage I read of a never ending journey that more than one person can take and anyone can choose to follow-on from. This journey might have come to an end for me now, but soon enough new roads will appear and after a short rest a new journey will begin again.

"The Road goes ever on and on
Out from the door where it began.
Now far ahead the Road has gone,
Let others follow it who can!
Let them a journey new begin,
But I at last with weary feet
Will turn towards the lighted inn,
My evening-rest and sleep to meet."

From “The Lord of the Rings”, J.R.R. Tolkien (1892-1973)

Bibliography

Abagyan RA, Batalov S (1997) Do aligned sequences share the same fold? J Mol Biol 273:355-368.

Adamczak R, Porollo A, Meller J (2004) Accurate prediction of solvent accessibility using neural networks-based regression. Proteins 56:753-767.

Albrecht M, Tosatto SC, Lengauer T, Valle G (2003) Simple consensus procedures are effective and sufficient in secondary structure prediction. Protein Eng 16:459-462.

Altschul SF (1989) Gap costs for multiple sequence alignment. J Theor Biol 138:297-309.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215:403-410.

Altschul SF, Koonin EV (1998) Iterated profile searches with PSI-BLAST--a tool for discovery in protein databases. Trends Biochem Sci 23:444-447.

Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389-3402.

An Y, Friesner RA (2002) A novel fold recognition method using composite predicted secondary structures. Proteins 48:352-366.

Bahr A, Thompson JD, Thierry JC, Poch O (2001) BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations. Nucleic Acids Res 29:323-326.

Bailey TL, Elkan C. (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. The second international conference on Intelligent Systems for Molecular Biology, 28-36.

Barker WC, Ketcham LK, Dayhoff MO (1978) A comprehensive examination of protein sequences for evidence of internal gene duplication. J. Mol. Evol. 10:265-281.

Barton GJ, Sternberg MJ (1987) Evaluation and improvements in the automatic alignment of protein sequences. Protein Eng 1:89-94.

Benner SA, Cohen MA, Gonnet GH (1993) Empirical and structural models for insertions and deletions in the divergent evolution of proteins. J Mol Biol 229:1065-1082.

Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The Protein Data Bank. Nucleic Acids Res 28:235-242.

Bishop CM (1995) Neural networks for pattern recognition. (eds). Clarendon Press; Oxford University Press: Oxford, New York, p xvii, 482.

Blanco FJ, Rivas G, Serrano L (1994) A short linear peptide that folds into a native stable beta-hairpin in aqueous solution. Nat Struct Biol 1:584-590.

Brenner SE, Koehl P, Levitt M (2000) The ASTRAL compendium for protein structure and sequence analysis. Nucleic Acids Res 28:254-256.

Bucher P, Karplus K, Moeri N, Hofmann K (1996) A flexible motif search technique based on generalized profiles. Comput Chem 20:3-23.

Bystroff C, Baker D (1998) Prediction of local structure in proteins using a library of sequence-structure motifs. J Mol Biol 281:565-577.

Bystroff C, Thorsson V, Baker D (2000) HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins. J Mol Biol 301:173-190.

Byvatov E, Schneider G (2003) Support vector machine applications in bioinformatics. Appl Bioinformatics 2:67-77.

Capriotti E, Fariselli P, Rossi I, Casadio R (2004) A Shannon entropy-based filter detects high- quality profile-profile alignments in searches for remote homologues. Proteins 54:351-360.

Carillo H, Lipman DJ (1988) The multiple sequence alignment problem in biology. SIAM J. Appl. Math. 48:1073-1082.

Carter P, Andersen CA, Rost B (2003) DSSPcont: Continuous secondary structure assignments for proteins. Nucleic Acids Res 31:3293-3295.

Chandonia JM, Karplus M (1999) New methods for accurate prediction of protein secondary structure. Proteins 35:293-306.

Chothia C, Lesk AM (1986) The relation between the divergence of sequence and structure in proteins. Embo J 5:823-826.

Chou PY, Fasman GD (1974) Prediction of protein conformation. Biochemistry 13:222-245.

Chung R, Yona G (2004) Protein family comparison using statistical models and predicted structural information. BMC Bioinformatics 5:183.

Churchill GA (1989) Stochastic models for heterogeneous DNA sequences. Bull Math Biol 51:79-94.

Corpet F (1988) Multiple sequence alignment with hierarchical clustering. Nucleic Acids Res 16:10881-10890.

Cregut D, Civera C, Macias MJ, Wallon G, Serrano L (1999) A tale of two secondary structure elements: when a beta-hairpin becomes an alpha-helix. J Mol Biol 292:389-401.

Cristianini N, Shawe-Taylor J (2000) An introduction to Support Vector Machines: and other kernel-based learning methods. (eds). Cambridge University Press: New York, p xiii, 189.

Crooks GE, Brenner SE (2004) Protein secondary structure: entropy, correlations and prediction. Bioinformatics 20:1603-1611.

Cuff JA, Barton GJ (1999) Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins 34:508-519.

Cuff JA, Barton GJ (2000) Application of multiple sequence alignment profiles to improve protein secondary structure prediction. Proteins 40:502-511.

Cuff JA, Clamp ME, Siddiqui AS, Finlay M, Barton GJ (1998) JPred: a consensus secondary structure prediction server. Bioinformatics 14:892-893.

Dayhoff M, Schwart R, Orcutt B (1978) A model of evolutionary change in proteins. In: (eds). Atlas of Protein Sequence and Structure. National Biomedical Research Foundation: Washington D.C., pp 345-352.

Dayhoff MO, Barker WC, Hunt LT (1983) Establishing homologies in protein sequences. Methods Enzymol 91:524-545.

de Bakker PI, Bateman A, Burke DF, Miguel RN, Mizuguchi K, Shi J, Shirai H, Blundell TL (2001) HOMSTRAD: adding sequence information to structure-based alignments of homologous protein families. Bioinformatics 17:748-749.

Derreumaux P (2001) Evidence that the 127-164 region of prion proteins has two equi-energetic conformations with beta or alpha features. Biophys J 81:1657-1665.

Dickerson RE, Timkovich R, Almassy RJ (1976) The cytochrome fold and the evolution of bacterial energy metabolism. J Mol Biol 100:473-491.

Do CB, Mahabhashyam MS, Brudno M, Batzoglou S (2005) ProbCons: Probabilistic consistency-based multiple sequence alignment. Genome Res 15:330-340.

Durbin R (1998) Biological sequence analysis: probalistic models of proteins and nucleic acids. (eds). Cambridge University Press: Cambridge, UK New York, p xi, 356.

Durbin R, Eddy S, Krogh A, Mitchison G (2000) Markov chains and hidden Markov models. In: (eds). Biological sequence analysis: probalistic models of proteins and nucleic acids. Cambridge University Press: Cambridge, UK New York, pp 46-79.

Ebedes J, Datta A (2004) Multiple sequence alignment in parallel on a workstation cluster. Bioinformatics 20:1193-1195.

Eck RV, Dayhoff MO (1966) Atlas of Protein Sequence and Structure 1966. (eds). National Biomedical Research Foundation: Silver Spring, Maryland, p

Eddy SR (1996) Hidden Markov models. Curr Opin Struct Biol 6:361-365.

Eddy SR (1998) Profile hidden Markov models. Bioinformatics 14:755-763.

Edgar RC (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 32:1792-1797.

Edgar RC, Sjolander K (2004a) COACH: profile-profile alignment of protein families using hidden Markov models. Bioinformatics 20:1309-1318.

Edgar RC, Sjolander K (2004b) A comparison of scoring functions for protein sequence profile alignment. Bioinformatics 20:1301-1308.

Eisenberg D, Schwarz E, Komaromy M, Wall R (1984) Analysis of membrane and surface protein sequences with the hydrophobic moment plot. J Mol Biol 179:125-142.

Felsenstein J (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 17:368-376.

Felsenstein J (1985) Confidence limits on phylogenies: an approach using the bootstrap. Evolution 39:783-791.

Feng DF, Doolittle RF (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J Mol Evol 25:351-360.

Fitch WM (1971) Toward defining the course of evolution: minimum change for a specific tree topology. Systematic Zoology 20:406-416.

Fitch WM, Margoliash E (1967) Construction of phylogenetic trees. Science 155:279-284.

Friedberg I, Kaplan T, Margalit H (2000) Evaluation of PSI-BLAST alignment accuracy in comparison to structural alignments. Protein Sci 9:2278-2284.

Frishman D, Argos P (1995) Knowledge-based protein secondary structure assignment. Proteins 23:566-579.

Frishman D, Argos P (1996) Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence. Protein Eng 9:133-142.

Frishman D, Argos P (1997) Seventy-five percent accuracy in protein secondary structure prediction. Proteins 27:329-335.

Garnier J, Gibrat JF, Robson B (1996) GOR method for predicting protein secondary structure from amino acid sequence. Methods Enzymol 266:540-553.

Garnier J, Osguthorpe DJ, Robson B (1978) Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. J Mol Biol 120:97-120.

Gibrat JF, Garnier J, Robson B (1987) Further developments of protein secondary structure prediction using information theory. New parameters and consideration of residue pairs. J Mol Biol 198:425-443.

Ginalski K, Pas J, Wyrwicz LS, von Grotthuss M, Bujnicki JM, Rychlewski L (2003) ORFeus: Detection of distant homology using sequence profiles and predicted secondary structure. Nucleic Acids Res 31:3804-3807.

Ginalski K, von Grotthuss M, Grishin NV, Rychlewski L (2004) Detecting distant homology with Meta-BASIC. Nucleic Acids Res 32:W576-581.

Gonnet GH, Cohen MA, Benner SA (1992) Exhaustive matching of the entire protein sequence database. Science 256:1443-1445.

Gotoh O (1986) Alignment of three biological sequences with an efficient traceback procedure. J Theor Biol 121:327-337.

Gotoh O (1993) Optimal alignment between groups of sequences and its application to multiple sequence alignment. Comput Appl Biosci 9:361-370.

Gotoh O (1996) Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. J Mol Biol 264:823-838.

Gribskov M, McLachlan AD, Eisenberg D (1987) Profile analysis: detection of distantly related proteins. Proc Natl Acad Sci U S A 84:4355-4358.

Gropp W, Lusk E, Doss N, Skjellum A (1996) A high-performance, portable implementation of the MPI Message-Passing Interface standard. Parallel computing 22:789-828.

Guermeur Y, Geourjon C, Gallinari P, Deleage G (1999) Improved performance in protein secondary structure prediction by inhomogeneous score combination. Bioinformatics 15:413-421.

Guo J, Chen H, Sun Z, Lin Y (2004) A novel method for protein secondary structure prediction using dual-layer SVM and profiles. Proteins 54:738-743.

Han KF, Baker D (1996) Global properties of the mapping between local amino acid sequence and local structure in proteins. Proc Natl Acad Sci U S A 93:5814-5818.

Haussler D, Krogh A, Mian I, Sjölander K. (1993). Protein modeling using hidden Markov models: Analysis of globins. International Conference on System Sciences, Hawaii, *1*, 792-802.

Henikoff S, Henikoff JG (1992) Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci U S A 89:10915-10919.

Henikoff S, Henikoff JG (1994) Position-based sequence weights. J Mol Biol 243:574-578.

Heringa J (1998) Detection of internal repeats: how common are they? Curr Opin Struct Biol 8:338-345.

Heringa J (1999) Two strategies for sequence comparison: profile-preprocessed and secondary structure-induced multiple alignment. Comput Chem 23:341-364.

Heringa J (2000) Computational methods for protein secondary structure prediction using multiple sequence alignments. Curr Protein Pept Sci 1:273-301.

Heringa J (2002) Local weighting schemes for protein multiple sequence alignment. Comput Chem 26:459-477.

Heringa J (personal communication)

Heringa J, Argos P (1991) Side-chain clusters in protein structures and their role in protein folding. J Mol Biol 220:151-171.

Heringa J, Taylor WR (1997) Three-dimensional domain duplication, swapping and stealing. Curr Opin Struct Biol 7:416-421.

Higgins DG, Sharp PM (1988) CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. Gene 73:237-244.

Hogeweg P, Hesper B (1984) The alignment of sets of sequences and the construction of phyletic trees: an integrated method. J Mol Evol 20:175-186.

Holm L, Sander C (1996) Mapping the protein universe. Science 273:595-603.

Hu WP, Kolinski A, Skolnick J (1997) Improved method for prediction of protein backbone U-turn positions and major secondary structural elements between U-turns. Proteins 29:443-460.

Hua S, Sun Z (2001) A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J Mol Biol 308:397-407.

Huang XQ, Hardison RC, Miller W (1990) A space-efficient algorithm for local similarities. Comput Appl Biosci 6:373-381.

Hubbard TJ, Ailey B, Brenner SE, Murzin AG, Chothia C (1998) SCOP, Structural Classification of Proteins database: applications to evaluation of the effectiveness of sequence alignment methods and statistics of protein structural data. Acta Crystallogr D Biol Crystallogr 54:1147-1154.

Jaroszewski L, Rychlewski L, Godzik A (2000) Improving the quality of twilight-zone alignments. Protein Sci 9:1487-1496.

Johnson MS, Doolittle RF (1986) A method for the simultaneous alignment of three or more amino acid sequences. J Mol Evol 23:267-278.

Jones DT (1999) Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 292:195-202.

Jones DT, Swindells MB (2002) Getting the most from PSI-BLAST. Trends Biochem Sci 27:161-164.

Jones DT, Taylor WR, Thornton JM (1992) The rapid generation of mutation data matrices from protein sequences. Comput. Appl. Biosci. 8:275-282.

Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22:2577-2637.

Karchin R, Cline M, Mandel-Gutfreund Y, Karplus K (2003) Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. Proteins 51:504-514.

Karplus K, Barrett C, Cline M, Diekhans M, Grate L, Hughey R (1999) Predicting protein structure using only sequence information. Proteins Suppl 3:121-125.

Karplus K, Barrett C, Hughey R (1998) Hidden Markov models for detecting remote protein homologies. Bioinformatics 14:846-856.

Karplus K, Hu B (2001) Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. Bioinformatics 17:713-720.

Karplus K, Karchin R, Barrett C, Tu S, Cline M, Diekhans M, Grate L, Casper J, Hughey R (2001) What is the value added by human intervention in protein structure prediction? Proteins Suppl 5:86-91.

Kim D, Xu D, Guo JT, Ellrott K, Xu Y (2003) PROSPECT II: protein structure prediction program for genome-scale applications. Protein Eng 16:641-650.

Kim H, Park H (2003) Protein secondary structure prediction based on an improved support vector machines approach. Protein Eng 16:553-560.

King RD, Ouali M, Strong AT, Aly A, Elmaghraby A, Kantardzic M, Page D (2000) Is it better to combine predictions? Protein Eng 13:15-19.

Kleinjung J, Douglas N, Heringa J (2002) Parallelized multiple alignment. Bioinformatics 18:1270-1271.

Kleinjung J, Romein J, Lin K, Heringa J (2004) Contact-based sequence alignment. Nucleic Acids Res. 32:2464-2473.

Kluge AG, Farris JS (1969) Quantitative phyletics and the evolution of anurans. Systematic Zoology 18:1-32.

Koh IY, Eyrich VA, Marti-Renom MA, Przybylski D, Madhusudhan MS, Eswar N, Grana O, Pazos F, Valencia A, Sali A, Rost B (2003) EVA: Evaluation of protein structure prediction servers. Nucleic Acids Res 31:3311-3315.

Kolinski A, Skolnick J, Godzik A, Hu WP (1997) A method for the prediction of surface "U"-turns and transglobular connections in small proteins. Proteins 27:290-308.

Krogh A, Brown M, Mian IS, Sjolander K, Haussler D (1994) Hidden Markov models in computational biology. Applications to protein modeling. J Mol Biol 235:1501-1531.

Krogh A, Riis SK (1999) Hidden neural networks. Neural Comput 11:541-563.

Lassmann T, Sonnhammer EL (2002) Quality assessment of multiple alignment programs. FEBS Lett 529:126-130.

Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science 262:208-214.

Lee C, Grasso C, Sharlow MF (2002) Multiple sequence alignment using partial order graphs. Bioinformatics 18:452-464.

Levin JM, Pascarella S, Argos P, Garnier J (1993) Quantification of secondary structure prediction improvement using multiple alignments. Protein Eng 6:849-854.

Lim VI (1974a) Algorithms for prediction of alpha-helical and beta-structural regions in globular proteins. J Mol Biol 88:873-894.

Lim VI (1974b) Structural principles of the globular organization of protein chains. A stereochemical theory of globular protein secondary structure. J Mol Biol 88:857-872.

Lin K, Kleinjung J, Taylor WR, Heringa J (2003) Testing homology with Contact Accepted mutatiOn (CAO): a contact-based Markov model of protein evolution. Comput Biol Chem 27:93-102.

Lin K, Simossis VA, Taylor WR, Heringa J (2005) A simple and fast secondary structure prediction method using hidden neural networks. Bioinformatics 21:152-159.

Lipman DJ, Altschul SF, Kececioglu JD (1989) A tool for multiple sequence alignment. Proc Natl Acad Sci U S A 86:4412-4415.

Liu JS, Lawrence CE (1999) Bayesian inference on biopolymer models. Bioinformatics 15:38-52.

Luisi DL, Wu WJ, Raleigh DP (1999) Conformational analysis of a set of peptides corresponding to the entire primary sequence of the N-terminal domain of the ribosomal protein L9: evidence for stable native-like secondary structure in the unfolded state. J Mol Biol 287:395-407.

Lüthy R, McLachlan AD, Eisenberg D (1991) Secondary structure-based profiles: use of structure-conserving scoring tables in searching protein sequence databases for structural similarities. Proteins 10:229-239.

Lüthy R, Xenarios I, Bucher P (1994) Improving the sensitivity of the sequence profile method. Protein Sci 3:139-146.

Macdonald JR, Johnson WC, Jr. (2001) Environmental features are important in determining protein secondary structure. Protein Sci 10:1172-1177.

Martelli PL, Fariselli P, Malaguti L, Casadio R (2002) Prediction of the disulfide bonding state of cysteines in proteins with hidden neural networks. Protein Eng 15:951-953.

McGuffin LJ, Jones DT (2003) Benchmarking secondary structure prediction for fold recognition. Proteins 52:166-175.

Mehta PK, Heringa J, Argos P (1995) A simple and fast approach to prediction of protein secondary structure from multiply aligned sequences with accuracy above 70%. Protein Sci 4:2517-2525.

Meiler J, Baker D (2003) Coupled prediction of protein secondary and tertiary structure. Proc Natl Acad Sci U S A 100:12105-12110.

Meiler J, Mueller M, Zeidler A, Schmaeschke F (2001) Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. Journal of Molecular Modelling 7:360-369.

Minor DL, Jr., Kim PS (1996) Context-dependent secondary structure formation of a designed protein sequence. Nature 380:730-734.

Minsky ML, Papert S (1988) Perceptrons: an introduction to computational geometry. (eds). MIT Press: Cambridge, Mass., p xv, 292.

Mittelman D, Sadreyev R, Grishin N (2003) Probabilistic scoring measures for profile-profile comparison yield more accurate short seed alignments. Bioinformatics 19:1531-1539.

Mizuguchi K, Deane CM, Blundell TL, Overington JP (1998) HOMSTRAD: a database of protein structure alignments for homologous families. Protein Sci 7:2469-2471.

Morgenstern B (1999) DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. Bioinformatics 15:211-218.

Muller T, Spang R, Vingron M (2002) Estimating amino acid substitution models: a comparison of Dayhoff's estimator, the resolvent approach and a maximum likelihood method. Mol. Biol. Evol. 19:8-13.

Muller T, Vingron M (2000) Modeling amino acid replacement. J. Comput. Biol. 7:761-776.

Murata M, Richardson JS, Sussman JL (1985) Simultaneous comparison of three protein sequences. Proc Natl Acad Sci U S A 82:3073-3077.

Murzin AG, Brenner SE, Hubbard T, Chothia C (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 247:536-540.

Nagano K (1973) Logical analysis of the mechanism of protein folding. I. Predictions of helices, loops and beta-structures from primary structure. J Mol Biol 75:401-420.

Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443-453.

Noble WS (2004) Support vector machine applications in computational biology. In: Schoelkopf B, Tsuda K, and Vert J-P (eds). Kernel methods in computational biology. MIT Press: Cambridge, Mass., pp 71-92.

Notredame C, Higgins DG, Heringa J (2000) T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol 302:205-217.

O'Sullivan O, Zehnder M, Higgins D, Bucher P, Grosdidier A, Notredame C (2003) APDB: a novel measure for benchmarking sequence alignment methods without reference alignments. Bioinformatics 19 Suppl 1:i215-221.

Ohlson T, Wallner B, Elofsson A (2004) Profile-profile methods provide improved fold-recognition: a study of different profile-profile alignment methods. Proteins 57:188-197.

Ortiz de Montellano PR (ed.) (1995). Cytochrome P450: structure, mechanism, and biochemistry. Plenum Press: New York.

Ouali M, King RD (2000) Cascaded multiple classifiers for secondary structure prediction. Protein Sci 9:1162-1176.

Pacheco PS (1997) Parallel programming with MPI. (eds). Morgan Kaufmann: San Francisco, Calif., p xxii, 418 p.

Pearson WR (2000) Flexible sequence similarity searching with the FASTA3 program package. Methods Mol Biol 132:185-219.

Pei J, Sadreyev R, Grishin NV (2003) PCMA: fast and accurate multiple sequence alignment based on profile consistency. Bioinformatics 19:427-428.

Petersen TN, Lundegaard C, Nielsen M, Bohr H, Bohr J, Brunak S, Gippert GP, Lund O (2000) Prediction of protein secondary structure at 80% accuracy. Proteins 41:17-20.

Pollastri G, Przybylski D, Rost B, Baldi P (2002) Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins 47:228-235.

Przybylski D, Rost B (2002) Alignments grow, secondary structure prediction improves. Proteins 46:197-205.

Qian N, Sejnowski TJ (1988) Predicting the secondary structure of globular proteins using neural network models. J Mol Biol 202:865-884.

Raghava GPS. (2002). APSSP2: A Combination Method for Protein Secondary Structure prediction Based on Neural Network and Example Based Learning. CASP 5.

Ramirez-Alvarado M, Serrano L, Blanco FJ (1997) Conformational analysis of peptides corresponding to all the secondary structure elements of protein L B1 domain: secondary structure propensities are not conserved in proteins with the same fold. Protein Sci 6:162-174.

Reymond MT, Merutka G, Dyson HJ, Wright PE (1997) Folding propensities of peptide fragments of myoglobin. Protein Sci 6:706-716.

Richards FM, Kundrot CE (1988) Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. Proteins 3:71-84.

Richardson JS, Richardson DC (1988) Amino acid preferences for specific locations at the ends of alpha helices. Science 240:1648-1652.

Rost B (1999) Twilight zone of protein sequence alignments. Protein Eng 12:85-94.

Rost B (2001) Review: protein secondary structure prediction continues to rise. J Struct Biol 134:204-218.

Rost B (personal communication)

Rost B, Sander C (1993) Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol 232:584-599.

Rost B, Sander C, Schneider R (1994) Redefining the goals of protein secondary structure prediction. J Mol Biol 235:13-26.

Rychlewski L, Jaroszewski L, Li W, Godzik A (2000) Comparison of sequence profiles. Strategies for structural predictions using sequence information. Protein Sci 9:232-241.

Sadreyev R, Grishin N (2003) COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. J Mol Biol 326:317-336.

Sadreyev RI, Baker D, Grishin NV (2003) Profile-profile comparisons by COMPASS predict intricate homologies between protein families. Protein Sci 12:2262-2272.

Saitou N (1990) Maximum likelihood methods. Methods Enzymol 183:584-598.

Saitou N, Nei M (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Biol Evol 4:406-425.

Sander C, Schneider R (1991) Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins 9:56-68.

Schaffer AA, Wolf YI, Ponting CP, Koonin EV, Aravind L, Altschul SF (1999) IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices. Bioinformatics 15:1000-1011.

Schiffer M, Edmundson AB (1967) Use of helical wheels to represent the structures of proteins and to identify segments with helical potential. Biophys J 7:121-135.

Schneider TD (2002) Consensus sequence Zen. Appl Bioinformatics 1:111-119.

Schoelkopf B, Tsuda K, Vert J-P (2004) Kernel methods in computational biology. Schoelkopf B, Tsuda K, and Vert J-P (eds). MIT Press: Cambridge, Mass., p ix, 400.

Schultz J, Copley RR, Doerks T, Ponting CP, Bork P (2000) SMART: a web-based tool for the study of genetically mobile domains. Nucleic Acids Res 28:231-234.

Schulz GE (1988) A critical evaluation of methods for prediction of protein secondary structures. Annu Rev Biophys Biophys Chem 17:1-21.

Selbig J, Mevissen T, Lengauer T (1999) Decision tree-based formation of consensus protein secondary structure prediction. Bioinformatics 15:1039-1046.

Serrano L, Fersht AR (1989) Capping and alpha-helix stability. Nature 342:296-299.

Sibbald PR, Argos P (1990) Weighting aligned protein or nucleic acid sequences to correct for unequal representation. J Mol Biol 216:813-818.

Simossis VA, Heringa J (2003) The PRALINE online server: optimising progressive multiple alignment on the web. Comput Biol Chem 27:511-519.

Simossis VA, Heringa J (2004a) The influence of gapped positions in multiple sequence alignments on secondary structure prediction methods. Comput Biol Chem 28:351-366.

Simossis VA, Heringa J (2004b) Integrating protein secondary structure prediction and multiple sequence alignment. Curr Protein Pept Sci 5:249-266.

Simossis VA, Heringa J (2005a) Improvement and limitations of secondary structure-guided multiple alignment quality. Bioinformatics (submitted).

Simossis VA, Heringa J (2005b) Local structure prediction of proteins. In: Xu Y, Xu D, and Liang J (eds). Computational methods for protein structure prediction and modeling. Springer Verlag, pp Chapter 8.

Simossis VA, Heringa J (2005c) PRALINE: a sequence alignment toolbox that integrates homology-extended and secondary structure information. Nucleic Acids Res 33:W1-W6.

Simossis VA, Heringa J (2005d) Sympred: optimal segmentation of consensus secondary structure prediction. BMC Bioinformatics (submitted).

Simossis VA, Kleinjung J, Heringa J (2003) An overview of Multiple Sequence Alignment. In: (eds). Current protocols in Bioinformatics. John Wiley: New York, pp 3.7.1-3.7.25.

Simossis VA, Kleinjung J, Heringa J (2005) Homology-extended sequence alignment. Nucleic Acids Res 33:816-824.

Sjolander K, Karplus K, Brown M, Hughey R, Krogh A, Mian IS, Haussler D (1996) Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. Comput Appl Biosci 12:327-345.

Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147:195-197.

Sobel E, Martinez HM (1986) A multiple sequence alignment program. Nucleic Acids Res 14:363-374.

Soding J (2004) Protein homology detection by HMM-HMM comparison. Bioinformatics. doi:10.1093/bioinformatics/bti125.

Sokal RR, Michener CD (1958) A statistical method for evaluating systematic relationships. University of Kansas Scientific Bulletin 28:1409-1438.

Stebbings LA, Mizuguchi K (2004) HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database. Nucleic Acids Res. 32 Database issue:D203-207.

Stoye J (1998) Multiple sequence alignment with the Divide-and-Conquer method. Gene 211:GC45-56.

Stoye J, Moulton V, Dress AW (1997) DCA: an efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. Comput Appl Biosci 13:625-626.

Sujatha S, Balaji S, Srinivasan N (2001) PALI: a database of alignments and phylogeny of homologous protein structures. Bioinformatics 17:375-376.

Tanner MA, Wong WH (1987) The calculation of posterior distributions by data augmentation. J. Am. Stat. Assoc. 82:528-550.

Taylor WR (1988) A flexible method to align large numbers of biological sequences. J Mol Evol 28:161-169.

Taylor WR (1998) Dynamic sequence databank searching with templates and multiple alignment. J. Mol. Biol. 280:375-406.

Taylor WR, Brown NP (1999) Iterated sequence databank search methods. Comput. Chem. 23:365-385.

Thompson JD, Gibson TJ, Plewniak F, Jeanmougin F, Higgins DG (1997) The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucleic Acids Res 25:4876-4882.

Thompson JD, Higgins DG, Gibson TJ (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res 22:4673-4680.

Thompson JD, Plewniak F, Poch O (1999a) BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics 15:87-88.

Thompson JD, Plewniak F, Poch O (1999b) A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Res 27:2682-2690.

Thompson JD, Plewniak F, Thierry J, Poch O (2000) DbClustal: rapid and reliable global multiple alignments of protein sequences detected by database searches. Nucleic Acids Res 28:2919-2926.

Tomii K, Akiyama Y (2004) FORTE: a profile-profile comparison tool for protein fold recognition. Bioinformatics 20:594-595.

Van Walle I, Lasters I, Wyns L (2004) Align-m--a new algorithm for multiple alignment of highly divergent sequences. Bioinformatics 20:1428-1435.

Vapnik VN (1995) The nature of statistical learning theory. (eds). Springer: New York, p xv, 188.

Vapnik VN (1998) Statistical learning theory. (eds). Wiley: New York, p xxiv, 736.

Vingron M, Argos P (1989) A fast and sensitive multiple sequence alignment algorithm. Comput Appl Biosci 5:115-121.

Vingron M, Sibbald PR (1993) Weighting in sequence space: a comparison of methods in terms of generalized sequences. Proc Natl Acad Sci U S A 90:8777-8781.

Viterbi AJ (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Informat. Theory IT-13:260-269.

von Ohsen N, Sommer I, Zimmer R (2003) Profile-profile alignment: a powerful tool for protein structure prediction. Pac. Symp. Biocomput. 252-263.

von Ohsen N, Sommer I, Zimmer R, Lengauer T (2004) Arby: automatic protein structure prediction using profile-profile alignment and confidence measures. Bioinformatics 20:2228-2235.

von Ohsen N, Zimmer R (2001) Improving profile-profile alignment via log-average scoring. In: Gascuel O and Moret B (eds). Algorithms in Bioinformatics. Spring-Verlag: Berlin Heidelberg NY, pp 11-26.

Wallace IM, O'Sullivan O, Higgins DG (2004) Evaluation of iterative alignment algorithms for multiple alignment. Bioinformatics

Wang G, Dunbrack RL, Jr. (2004) Scoring profile-to-profile sequence alignments. Protein Sci 13:1612-1626.

Wang L, Jiang T (1994) On the complexity of multiple sequence alignment. J Comput Biol 1:337-348.

Ward JJ, McGuffin LJ, Buxton BF, Jones DT (2003) Secondary structure prediction with support vector machines. Bioinformatics 19:1650-1655.

Waterman MS (1986) Multiple sequence alignment by consensus. Nucleic Acids Res 14:9095-9102.

Waterman MS, Eggert M (1987) A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. J Mol Biol 197:723-728.

Waterman MS, Jones R (1990) Consensus methods for DNA and protein sequence alignment. Methods Enzymol 183:221-237.

Yona G, Levitt M (2002) Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. J Mol Biol 315:1257-1275.

Yu YK, Wootton JC, Altschul SF (2003) The compositional adjustment of amino acid substitution matrices. Proc. Natl. Acad. Sci. U.S.A. 100:15688-15693.

Zemla A, Venclovas C, Fidelis K, Rost B (1999) A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. Proteins 34:220-223.

Zhang Z, Lindstam M, Unge J, Peterson C, Lu G (2003) Potential for dramatic improvement in sequence alignment against structures of remote homologous proteins by extracting structural information from multiple structure alignment. J Mol Biol 332:127-142.

Zhu J, Liu JS, Lawrence CE (1998) Bayesian adaptive sequence alignment algorithms. Bioinformatics 14:25-39.

Zvelebil MJ, Barton GJ, Taylor WR, Sternberg MJ (1987) Prediction of protein secondary structure and active sites using the alignment of homologous sequences. J Mol Biol 195:957-961.

Samenvatting

Het ontwikkelen van kennis over eiwitfuncties is één van de belangrijkste aspecten van de moleculaire biologie. Deze kennis is van groot belang om te begrijpen hoe een organisme werkt en is essentieel voor het ontwikkelen van strategieën voor het voorkomen van ziekten and het ontwikkelen van behandelwijzen. De functie van een eiwit is een gevolg van de tertiaire structuur en de specifieke functionele modules die deze structuur bevat. Zo leidt de beschikbaarheid van een correcte 3-dimensionale structuur meestal tot betrouwbare conclusies over wat een gegeven eiwit doet en soms zelfs over de manier waarop het dat doet.

Alhoewel veel voortgang is geboekt in het experimenteel oplossen van de 3-dimensionale structuur van eiwitten, loopt het tempo hiervan nog steeds ver achter bij de snelheid waarmee wereldwijd in meer dan honderd genoomprojecten de DNA sequenties van vele organismen opgelost worden. Hierdoor, maar zeker ook door het beschikbaar komen van de complete *first draft* van het menselijk genoom in april 2003, zijn methodes om sequentie-informatie te analyseren, en om het gat tussen sequentiële en structurele data te dichten, zeer in de belangstelling geraakt. De meest gebruikte manier om ideeën over de structuur en functie van een eiwitsequentie op te doen, is het vaststellen van overeenkomsten tussen eiwitsequenties. Dit kan gedaan worden door relaties vast te stellen tussen een onbekend eiwit en een eiwit waarvan de structuur en/of functie bekend is, of tussen regio's van deze eiwitsequenties die belangrijk lijken te zijn op grond van evolutionaire conserveringspatronen. Een probleem hierbij is dat het vergelijken van sequenties zeer veel moeilijker wordt naarmate de sequenties een grotere evolutionaire afstand hebben. Het kan zelfs zo zijn dat de overeenkomst tussen twee evolutionair verwante sequenties helemaal niet meer is vast te stellen. Dit

proefschrift gaat in op het bovenstaande probleem en beschrijft een aantal nieuwe methoden voor het vergelijken van divergente sequenties, gebaseerd op integratie van secundaire structuurvoorspelling en uitbreiding van de evolutionaire informatie op iedere positie van een gegeven sequentie.

Na een algehele introductie wordt in hoofdstuk 2 en 3 een overzicht gegeven van sequence alignment en van secundaire structuurvoorspelling.

Hoofdstuk 4 behandelt vervolgens het gebruik van evolutionaire informatie in bestaande methoden voor het voorspellen van de secundaire structuur. Een nieuw algoritme wordt gepresenteerd voor het combineren van deze voorspellingen in een optimaal gesegmenteerde consensusvoorspelling. Daarnaast wordt een strategie beschreven om informatie die door de meeste moderne methoden wordt verwijderd voordat de secundaire structuur wordt voorspeld, te herintroduceren. Er wordt aangetoond dat deze aanpak voor een aantal methoden de voorspellingskwaliteit verbetert.

In hoofdstuk 5 wordt een nieuwe methode voor secundaire structuurvoorspelling geïntroduceerd die is gebaseerd op een zogenaamd “hidden neural network”. Het neurale netwerk is getraind om de input aminozuren in 7 klassen van secundaire structuren in te delen in plaats van in 3 klassen, zoals meestal gebeurd, waarna een hidden Markov model de output van het neurale netwerk optimaliseert. Vergeleken met de beste alternatieve methoden, combineert de methode combineert efficiëntie met een superieure kwaliteit voor het voorspellen van β-strands. Deze verbetering is belangrijk omdat de β-strand de moeilijkste secundaire structuur is om te voorspellen.

In hoofdstuk 6 wordt een nieuwe globale multiple alignment methode gepresenteerd die gebruik maakt van profiles waarin de evolutionaire informatie is uitgebreid op grond van homology searching. Het algoritme gebruikt de zo gevormde “homology extended profiles” in plaats van de oorspronkelijke input-sequenties. De kwaliteit van de methode wordt vergeleken met andere alignment methoden, waarbij gebruik wordt gemaakt van een standaardset van structurele alignments als referentie. Uit deze vergelijking blijkt dat de nieuwe methode de beste resultaten geeft en speciaal geschikt is voor het maken van alignments van sequenties die meer dan 70% gedivergeerd zijn.

Hoofdstuk 7 beschrijft een methode waarin de strategie voor multiple alignment, gepresenteerd in het voorgaande hoofdstuk, wordt geïntegreerd met secundaire structuurvoorspelling. Het niveau van de verbetering in de alignments door het integreren van voorspelde secundaire structuren wordt vergeleken met de verbetering die verkregen wordt wanneer “juiste”, d.w.z. geobserveerde secundaire structuren worden gebruikt. Hierbij worden bepaalde systematische fouten, gemaakt door de meeste voorspellingsmethoden, geïdentificeerd als belangrijkste beperkende factoren.

In hoofdstuk 8 wordt de online server van het alignment programma PRALINE beschreven. De beschikbare interfaces en opties van het programma, en de verschillende mogelijkheden om een verkregen alignment weer te geven, worden behandeld.

Tenslotte wordt in hoofdstuk 9 een algehele discussie gepresenteerd, waarin de belangrijkste onderzoeksvragen worden samengevat en een aantal mogelijkheden tot verdergaand onderzoek wordt behandeld.

Summary

The understanding of protein function is a fundamental area of molecular biology. It provides essential insight into how an organism works and more importantly helps with the study of disease and the development of preventative strategies or cures. The function of a protein is mostly determined by its (tertiary) structure and the specific functional units it comprises. As a result, a correct representation of a proteins structure allows for confident conclusions to be made about what a protein does and sometimes how it does it. At present, although great progress has been made in determining the three-dimensional structure of proteins, the process is considerably slower than the rate at which sequence information is generated by the over one hundred genome sequencing projects currently underway. More importantly, a first draft of the complete human genome has been available since April 2003 and therefore the improvement of methods that analyse this information and close the gap between sequence and structure is more important than ever. The most essential method for extrapolating structure and function information from sequence is the identification of similarities between proteins, either by comparing unknown proteins to well characterised ones or through the identification of regions that are essential for a protein's function based on their level of evolutionary conservation. However, accurate comparison and matching of these regions becomes very hard when very distant sequences are involved and the information if often undetectable. The work described in this thesis deals with ways in which the quality and reliability of these comparisons can be improved by integrating predicted secondary structure and additional position-specific evolutionary information.

In Chapters 2 and 3 an overview of sequence alignment and secondary structure prediction theory are presented, respectively.

In Chapter 4 a study of the current use of evolutionary information in secondary structure prediction is described. In addition, a new algorithm for combining information into an optimally segmented consensus is presented. The study shows that many state-of-the-art secondary structure prediction methods remove essential information before performing a prediction that when re-introduced improves the quality of the predictions for some methods.

In Chapter 5 the implementation of a hidden neural network for secondary structure prediction is described. The neural network is trained to classify the input amino acids into seven classes instead of three and the hidden Markov model optimises the network output. The method combines speed and a superior strand prediction accuracy, which is the hardest secondary structure type to predict, compared to several state-of-the-art prediction methods.

In Chapter 6 an algorithm for the global multiple alignment of sequences using profiles containing extended evolutionary information collected from database searching is presented. The algorithm performs a multiple alignment using the progressive strategy using the homology-extended profiles instead of the given sequences. The quality of the alignments it generates is compared to state-of-the-art multiple alignment methods using a standard set of structural alignments as the standard of truth. The highest improvement is achieved when aligning very hard alignment cases with sequences that have less than 30% sequence identity.

In Chapter 7, the integration of predicted secondary structure into the multiple alignment strategy presented in the previous chapter and a standard alignment method is described. The level of possible improvement is compared to that when using “true” structural information and the limiting factors are identified to be systematic prediction errors that are commonly made by most state-of-the-art prediction methods.

In Chapter 8 the online server of the alignment toolbox PRALINE is described. The web-based interfaces, program options and alignment representation details are discussed.

In Chapter 9 the main research points are summarised and future possibilities that spawn from this work are discussed.

Victor A. Simossis graduated from the University of Edinburgh with a BSc Honours degree in Molecular Biology in 1999. Until 2001 he worked on the characterisation of *poly*, a novel gene in *Drosophila melanogaster* in the Wellcome Institute for Cell and Molecular Biology (ICMB) of the University of Edinburgh, for which he received an MSc by research with distinction in Cell Biology. He immediately started a PhD in the MRC National Institute for Medical Research (NIMR) in London and in 2002 transferred together with his supervisor to the Vrije Universiteit (VU) Amsterdam to complete his research. Since March 2005 he has returned to Greece where he is soon to serve his military service.